

**United States Military Academy**

**West Point, New York 10996**

## **Recruiter Selection Model**

**OPERATIONS RESEARCH CENTER OF EXCELLENCE**

**TECHNICAL REPORT NO: DSE-TR-0623**

**DTIC #: ADA443651**

Lead Analyst and Senior Investigator

**Lieutenant Colonel John Brantley Halstead, Ph.D.**

Assistant Professor, Operations Research Center of Excellence

Directed by

**Lieutenant Colonel Simon Goerger, Ph.D.**

Director, Operations Research Center of Excellence

Approved by

**Colonel Michael L. McGinnis, Ph.D.**

Professor and Head, Department of Systems Engineering

**May / 2006**

**Distribution A: Approved for public release; distribution is unlimited.**

**20060314 016**

# **Recruiter Selection Model**

Lead Analyst

**Lieutenant Colonel John Brantley Halstead, Ph.D.**

Assistant Professor, Operations Research Center of Excellence

Senior Investigator

**Lieutenant Colonel John Brantley Halstead, Ph.D.**

Assistant Professor, Department of Systems Engineering

**OPERATIONS RESEARCH CENTER OF EXCELLENCE**

**TECHNICAL REPORT NO: DSE-TR-0623**

**DTIC #: ADA443651**

Directed by

**Lieutenant Colonel Simon Goerger, Ph.D.**

Director, Operations Research Center of Excellence

Approved by

**Colonel Michael L. McGinnis, Ph.D.**

Professor and Head, Department of Systems Engineering

**May 2006**

The Operations Research Center of Excellence is supported by the Assistant Secretary of the Army  
(Financial Management & Comptroller)

This Research was sponsored by: The United States Army Recruiting Command and the Assistant Secretary of the  
Army for Manpower and Reserve Affairs

**Distribution A: Approved for public release; distribution is unlimited.**

## **Abstract**

This research enhances Industrial and Organizational Psychology (IO) by providing a statistical prediction of job performance derived from psychological inventories and biographical data. The research uses a combination of statistical learning, feature selection methods, and multivariate statistics to determine the better prediction function approximation with features obtained from the Non Commissioned Officer Leadership Skills Inventory (NLSI) and biographical data. The research created a methodology for iteratively developing a statistical learning model. After exploring RandomForest, Support Vector Regression, and Linear Regression, the RandomForest model best predicts recruiter performance for these data. The model's performance was further enhanced by using a greedy feature selection method to determine the best subset of features that produced the best model generalization. The resulting model runs in R statistical language and is controlled within an Excel worksheet environment by using Visual Basic Application (VBA) language to call R. The end product enables general user utilization of a statistically eloquent model, normally reserved for advanced researchers, engineers, statisticians, and economists. The model represents a multi-modal relationship primarily between recruiter age and NLSI score and to a lesser degree, 34 other features.

This study is a result of Army Recruiting Initiatives earlier research into the feasibility of constructing a recruiter prediction model. The model, because of Excel deployment, is convenient to use. The convenience facilitates the model's migration from Center One to, perhaps, TRADOC. The model also provides significant cost benefits to the Army. If the model is used, recruiting potentially receives those individuals with the inherent skill sets for recruiting. Equipping recruiting with the right individual should reduce the number of required recruiters because the command's gross write rate should increase. This also retains more junior leadership in the operational army, reducing the personal turnover associated with placing a good leader, without inherent recruiting skills, within recruiting.

## About the Authors

Lieutenant Colonel John B. Halstead is a 1986 graduate of the United States Military Academy, West Point, NY. He has a BS from USMA in mathematics of operations research. Other degrees include a MS from Kansas State University, Manhattan, KS in operations research (1997) and a Ph.D. from the University of Virginia, Charlottesville, VA in systems and information engineering (2005).

John currently serves as an Assistant Professor within the Department of Systems Engineering, United States Military Academy at West Point, NY. He is an active duty Army Officer with a balance of operational leadership positions and senior staff positions. More recent duties included Strategic Planning Officer for the United States Army Accessions Command and Recruiting Command, Strategic Concepts Officer for United States Army Recruiting Command, and a Maneuver Observer Controller. Current and previous research interests include feature selection, statistical learning, multivariate statistics, market research, and classification. He may be contacted at [john.halstead@usma.edu](mailto:john.halstead@usma.edu).

## Acknowledgements

I sincerely appreciate the collaboration, insights, and support of Dr. Linda Ross and Ms. Birgit Valdez from Recruiting Command's Center One. I further thank Dr. Len White, Army Research Institute, for his expertise and collaboration. I am especially thankful for Dr. Wally Borman, the predominate expert in IO psychology, for his gentleman guidance and his sincere contribution to the armed forces of the United States. I also express great appreciation for his employees at Personnel Decision Research Incorporated, especially Ms. Valentina Lee who often and enthusiastically provided me with much needed test and numerical data and other information. From within the Systems Engineering Department, I thank LTC Dale Henderson, Ph.D. for his relevant insight into merging research into human resource applications. From the Social Sciences Department, I thank and appreciate MAJ Kyle Jette's economic input.

# Table of Contents

<b>Abstract.....</b>	<b>iii</b>
<b>About the Authors.....</b>	<b>iv</b>
<b>Acknowledgements .....</b>	<b>iv</b>
<b>Table of Contents .....</b>	<b>v</b>
<b>List of Figures.....</b>	<b>viii</b>
<b>List of Tables .....</b>	<b>x</b>
<b>Chapter 1: Introduction .....</b>	<b>1</b>
1.1 Army Recruiting Initiatives .....	1
1.2 Personnel Decisions Research Institutes' (PDRI) Development of the Non- Commissioned Officer Leadership Skills Inventory (NLSI). ....	1
1.3 Statistical Learning and Prediction .....	2
<b>Chapter 2: Problem and Vision.....</b>	<b>5</b>
2.1 Problem Statement .....	5
2.2 Vision.....	6
<b>Chapter 3: Modeling Methodology and Data Structures.....</b>	<b>8</b>
3.1 Modeling Methodology .....	8
3.1.1. Methodology Overview .....	8
3.1.2. Conjecture.....	9
3.1.3. Feature Selection .....	9
3.1.4. Statistical Learning.....	10
3.1.4.1 Linear Regression .....	10
3.1.4.2 Tree Regression .....	11
3.1.4.3 Random Forest.....	13
3.1.4.4 Support Vector Regression .....	13

3.1.5.	Analysis .....	15
3.1.6.	Model Refinement .....	16
3.2	Data .....	16
3.2.1.	Response Variable .....	16
3.2.2.	Prediction Variables (Features) .....	17
3.2.2.1	Data Cleansing .....	17
3.2.2.2	Training and Validation Data.....	17
<b>Chapter 4:</b>	<b>Modeling Methodology Results.....</b>	<b>18</b>
4.1	Initial Data Understanding .....	18
4.2	Initial Feature Selection .....	18
4.3	Initial Statistical Learning.....	20
4.4	Initial Analysis .....	23
4.5	Refinements .....	25
<b>Chapter 5:</b>	<b>Model .....</b>	<b>28</b>
5.1	Model Specifications .....	28
5.1.1.	Prediction Model .....	28
5.1.2.	Recruiter Order of Merit Ranking and Predicted Gross Write Rate.....	36
5.2	Data Structures .....	40
5.2.1.	Model Prediction Data .....	40
5.2.2.	Model Implementation and Updates.....	41
<b>Chapter 6:</b>	<b>Conclusions .....</b>	<b>42</b>
6.1	Research Contributions .....	42
6.2	Recommended Model Deployment .....	42
6.3	Estimated Return on Investment .....	42
<b>Bibliography</b>	<b>.....</b>	<b>44</b>
<b>Appendix A: List of Abbreviations</b>	<b>.....</b>	<b>46</b>
<b>Appendix B: Data Definitions</b>	<b>.....</b>	<b>47</b>
<b>Appendix C: R Code</b>	<b>.....</b>	<b>53</b>

<b>Appendix D: RExcel Random Forest Model.....</b>	<b>62</b>
<b>Distribution List.....</b>	<b>66</b>
<b>REPORT DOCUMENTATION PAGE – SF298 .....</b>	<b>67</b>

## List of Figures

Figure 1, Correlation Plot between NLSI Score and Gross Write Rate (Corr = 0.1724) .....	6
Figure 2, Modeling by Statistical Learning Methods: The Process of Conjecture, Feature Selection, Statistical Learning, and Analysis.....	8
Figure 3, Random Forest Model, Predicted Values Are Green and Actual Observations Are Black .....	21
Figure 4, Support Vector Regression Model 1 (SVR 1), Predicted Values Are Blue and Actual Observations .....	22
Figure 5, Support Vector Regression Model 2 (SVR 2), Predicted Values Are Red and Actual Observations Are Black .....	22
Figure 6, Linear Regression Model, Predicted Values Are Purple and Actual Observations Are Black .....	23
Figure 7, Random Forest Error Rates as Related to the Number of Trees Grown within the Forest .....	24
Figure 8, Top 10 Random Forest Features.....	24
Figure 9, Pairwise Scatter Plots of Age, NLSI Score, and Gross Write Rate.....	25
Figure 10, Random Forest Best Subset Selection as Determined by Minimum MSE .....	26
Figure 11, Improved Random Forest Model via Feature Selection from an Ordered Greedy Algorithm.....	27
Figure 12, Random Forest Important Model Features.....	27
Figure 13, U.S. Army Recruiter Selection Model Window.....	29
Figure 14, About the Model Window .....	29
Figure 15, Comprehensive R Archive Network (CRAN) Site .....	30
Figure 16, R for Windows Site .....	31
Figure 17, R-2.2.1 for Windows Site.....	31
Figure 18, File Download Window .....	31
Figure 19, Other Software on CRAN Site .....	32
Figure 20, Index of RExcel Software Site .....	32
Figure 21, Setup R (D)COM Server Window .....	33
Figure 22, RGui Package Installation Window .....	34



Figure 23, CRAN Mirror Window.....	34
Figure 24, Packages Window .....	35
Figure 25, Recruiter Selection Model Window .....	35
Figure 26, Response Surface of Predicted Gross Write Rate as a Function of NLSI Score and Age.....	37
Figure 27, Gross Annual Contracts Distributed by Order of Merit List (Individual Contracts are the solid line and Accumulative Contracts are the dashed lines).....	38
Figure 28, Histogram of Model's Output, Predicted Gross Write Rate .....	39
Figure 29, Return on Investment by Implementing NLSI and the Recruiter Selection Model	43

## List of Tables

Table 1, Ordered Step-wise Feature Selection Results (Ordered from Best to Better Features)	19
Table 2, Ordered Random Forest Feature Selection Results (Ordered from Best to Better)..	20
Table 3, Model MSE Comparison .....	23
Table 4, Recruiter Order of Merit List Example.....	40

## **Chapter 1: Introduction**

### ***1.1 Army Recruiting Initiatives***

In 2001, the Secretary of the Army, the Honorable Mr. Louis Caldera, created and resourced numerous recruiting initiatives (Army Recruiting Initiatives) designed to propel Army recruiting by modernizing the sales force, matching programs and incentives to the current youth market, and reducing barriers to Army enlistment. Part of the Army Recruiting Initiatives was the creation of a recruiter selection instrument. The initial intent of the recruiter selection instrument was to create and implement a survey instrument that would accurately predict a noncommissioned officer's (NCO) potential recruiting performance. Placed within the NCO education system (NCOES), the instrument would survey the NCO population. Using a predicted performance evaluation, Training and Doctrine Command (TRADOC) could establish a recruiter order of merit list that facilitated Human Resource Command's (HRC) recruiter selection.

The returns on investment are significant to the Army and Army recruiting. By choosing those NCOs with inherent skill sets for recruiting, Army recruiting obtains NCOs who should perform better than current detailed recruiters. As a result of improved performance, the United States Army Recruiting Command (USAREC) increases its average gross write rate (GWR), a metric measuring the total contracts a single recruiter obtains in one month. Given a contract mission, an increased GWR decreases the number of required recruiters. Hence, USAREC reduces its recruiter population and the Army retains more NCOs in operation formations. Second order effects created by detailing good NCOs lacking recruiting skills are also reduced. These effects include personnel turmoil from failing Recruiting and Retention School (RRS) and pre-mature cause for relief within USAREC as well as the potentially harmful career damage to capable operational leaders.

### ***1.2 Personnel Decisions Research Institutes' (PDRI) Development of the Non-Commissioned Officer Leadership Skills Inventory (NLSI).***

In conjunction with Army Research Institute (ARI), PDRI developed the NLSI to measure potential recruiter success. The NLSI is a psychological survey instrument that measures attributes and skills believed to be relevant to recruiter performance. Both ARI and PDRI have

distinguished histories in analyzing and developing industrial and organizational (IO) psychology instruments. Dr. Walter Borman, the CEO of PDRI, is one of the major IO psychology field leaders and is personally responsible for much of the current IO body of knowledge associated with developing and analyzing IO psychological inventories (Borman and Rosse, 1980) (PDRI, 2005).

The physical construction of the NLSI consists of three parts and takes approximately 60 to 90 minutes to complete. The first two parts measure attributes related to recruiter performance. Examples include work orientation, leadership, and interpersonal skills. The third part measures situational judgment skills such as sales skill, social judgment, and leadership. The questions are designed within an elaborate process that begins with a job analysis. The NLSI instrument produces an NLSI score, which may have predictive potential for estimating recruiter productivity.

The Army began testing NCOs detailed to recruiting at the Army Recruiting Course (ARC) at the Recruiting and Retention School (RRS) beginning in January 2002. Initially, the NLSI was given on paper. In January 2003, the NLSI began computerized administration. Over 4,000 detailed recruiters have been tested with the NLSI. Because of these historical records, USAREC can pair the NLSI results with the recruiter's average GWR for each record. The mapping of a GWR with the NLSI score and components permits the application of information technologies such as statistical learning. The situation provides an opportunity to leverage recent gains within statistical learning theory and data mining to learn relationships between the NLSI score, NLSI components, and GWR. In application, we determine a function approximation that best models these relationships. Ultimately, the function approximation predicts potential recruiter GWR.

### ***1.3 Statistical Learning and Prediction***

Statistical learning has increased its value by providing a relevant role within science, finance, and industry. The goal of statistical learning is prediction or classification. In statistical learning, we learn from the data. This approach is different from traditional statistics, where a hypothesis is accepted or rejected based upon the result of a test statistic. Statistical learning facilitates the data to speak, which exposes relationships within the data and provides a prediction or classification.

Typically, there exists some outcome measurement that is quantitative or categorical. An example of a quantitative outcome is a commodity price, while a categorical response example is disease or no disease. We call the outcome a response. In classical examples, the outcome has been labeled the dependent variable. When we speak of categorical responses, we often desire to predict the class of the response, which is called classification. When the response is quantitative, we use regression to predict the response. The prediction occurs from a set of features, which are traditionally labeled prediction variables or independent variables. The pairing of features and the outcome measurement form a training set of data. By using the training data, we are able to construct prediction models that learn from the data. These models permit us to predict the outcome of new unseen data. Good learning models permit accurate prediction, which is often measured in classification accuracy or, for regression, some error metric.

Supervised learning occurs when we use a training data set of features and responses to learn the data and, on unseen data, use the features to predict a response. Essential to supervised learning is the pairing of features with a response. The artificial intelligence body of knowledge uses the pairing within a learning algorithm designed to minimize the errors between a predicted and actual response. The process is generally referred to as learning by example. Somewhat less glamorous is the approach taken within applied mathematics and statistics, which uses function approximation and estimation. Function approximation encourages learning with geometrical concepts of Euclidian spaces and probabilistic inferences. Some examples of function approximations for regression include CART Trees, Support Vector Regression, Multivariable Regression, Random Forest, and Artificial Neural Networks. Function approximations are typically developed by minimizing sum of square errors or through maximum likelihood estimation. Some function approximations are developed with more sophisticated methods, such as support vector regression using Lagrange multipliers. Hastie (2002) is one predominant industry standard for statistical learning.

This particular situation lends itself to two major contributions. The first is the application of statistical learning with a purpose of accurately predicting potential recruiter gross write rate for recruiter selection. The second contribution is the introduction of statistical learning into IO psychology, which links job performance instruments with a prediction of job performance. The first contribution potentially creates large returns on investment for the Army. Recruiting

obtains the right talent for detailed recruiters. The Army retains more junior and career NCO leaders within its operational formations. The second contribution is a new development within a rich body of IO psychology knowledge that transforms IO psychological analysis into performance prediction.

The approach is different from traditional statistics. Rather than hypothesizing a result and using a test statistic to confirm or deny the hypothesis, statistical learning tests various function approximations to find the best model for predicting gross write rate. The data pairs  $\{x_i, y_i\}$  are points in a  $(p+1)$  dimensional Euclidean space. A function  $f(x)$  has a domain equivalent to  $p$  dimensional input subspace and is related to a model by  $y_i = f(x_i) + \varepsilon_i$ . There are two goals: the first is to determine the best approximation (the best model) for  $f(x)$  and the second finds the better data pairs  $\{x_i, y_i\}$  for that model. The first goal develops the most appropriate model for the data. The second goal conducts feature selection. Feature selection ultimately increases the generalization of the model. Generalization is the model's ability to accurately classify or predict new unseen observations. Better generalization is achieved by removing noise features from the data pairs so that only important features that contribute information to the model remain. Within this application, the data pairs are defined by the features and the response. Our initial features (prior to feature selection) are the NLSI score and the NLSI components, while our response is gross write rate.

## **Chapter 2: Problem and Vision**

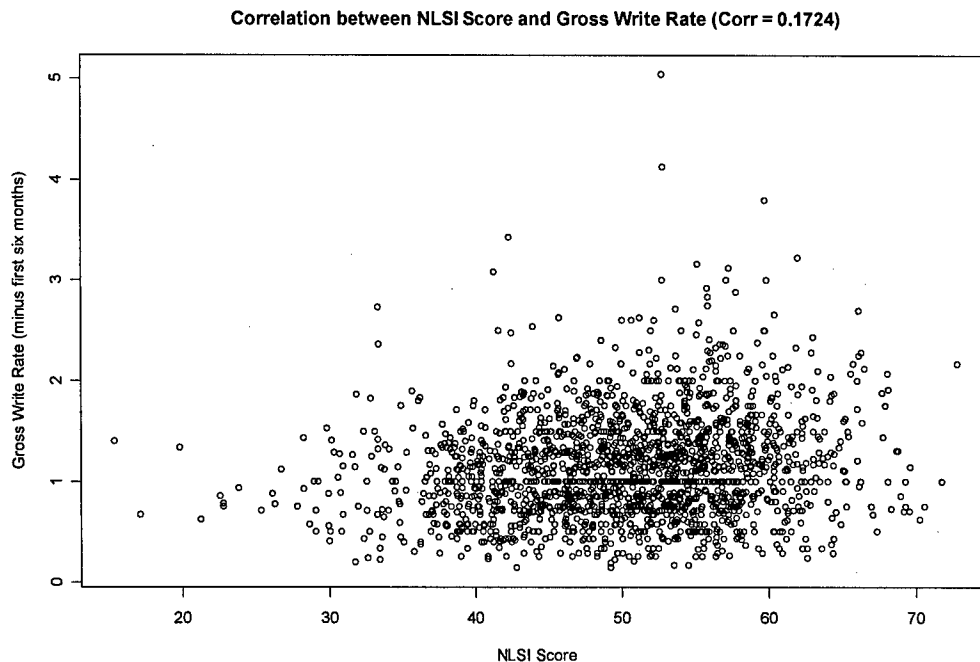
### **2.1 Problem Statement**

The Army's recruiting force is comprised of two populations: detailed recruiters and career recruiters (79R). Career recruiters are either recruited from the operational army or are selected from those detailed recruiters who, through their recruiting success, volunteer for the duty. For the most part, career recruiters have the desire to recruit and inherently possess the right skill sets that enables successful recruiting. The challenge exists with the detailed recruiter force.

Detailed recruiters are chosen from the operational army. Prior to recruiting detail, these NCOs were successful within their military occupational skill (MOS), providing essential leadership at the tactical level. The majority of detailed recruiters are drawn from the combat arms and the remaining is selected from the myriad of other specialties. Whether their leadership was used in combat, combat support, or combat service support role, few relied upon sales skills for their success. The skills required of an infantry squad leader, a tank commander, and an intelligence analyst are starkly different than those skills required of a recruiter. When a detailed recruiter doesn't have inherent recruiting skills, all involved parties are some what damaged. Examples of immediate effects are: 1) the Army temporarily loses essential junior leaders in its operational formations; 2) recruiting effectiveness decreases; and 3) the individual NCO is potentially damaged by frequent rejection, failure, and a decline in their primary military occupational skill set. A critic may claim the RRS and the recruiting station can train all NCOs to become successful recruiters. Yet, in all the years of the All Volunteer Force (AVF), this statement has not been fulfilled.

The problem centers on selecting those personnel from the operational army who have the inherent skill set for recruiting. The NLSI is the Army's initial foray addressing the problem. From a human factors perspective, the NLSI does have a reasonably good correlation to GWR. The correlation is stated and graphically depicted in Figure 1. The average GWR excludes the first six months of production so that a learning curve is eliminated from the response, reflecting a better indication of recruiter production (PDRI, 2005). The cone like shape of the scatter plot suggests a non-linear relationship exists between the feature (NLSI Score) and the response (GWR). The plot demonstrates low NLSI scores predict the likelihood of recruiting challenges. The plot substantiates PDRI analysis (PDRI, 2005). However, for most of the higher NLSI

scores, the score alone may not adequately predict potential production. Although there is an increase in GWR as NLSI increases, there remains a sizable population at or below a GWR equivalent to one. The NLSI is a sound instrument and should remain the foundation for a recruiter selection instrument. Improving the NLSI prediction involves the discovery of other features that contribute information to the NLSI score for predicting gross write rate and the discovery of a function approximation that more accurately uses those features to predict gross writer rate.



**Figure 1, Correlation Plot between NLSI Score and Gross Write Rate (Corr = 0.1724)**

## **2.2 Vision**

The vision is to select individuals with the inherent skill sets for recruiting from the operational army into detailed recruiting. Accomplishing the vision relies upon discovering a model that accurately predicts potential recruiter gross write rate and administering the model at the right time and place within the Army.

This research enhances the industrial and organization psychology body of knowledge by incorporating statistical learning theory. The addition of statistical learning theory permits the best possible prediction of recruiter performance. The statistical learning modeling and the



model results are discussed in chapters 3, 4, and 5. Model deployment is discussed in the conclusion.

## Chapter 3: Modeling Methodology and Data Structures

### 3.1 Modeling Methodology

#### 3.1.1. Methodology Overview

Because statistical learning is exploratory in nature, we developed a system that facilitates the discovery of the best function approximation paired with the best set of model features. The system performs an iterative process until the best possible function approximation and feature set can be obtained from the data. During the process, the system also has an advantage of providing information and intelligence on its current state. The information and intelligence can result from any of the sub-processes. Intelligence is different than information in that it provides analysis of the information. An example of information is a list of important variables. Intelligence, in this example, is an ordered list of variables and a metric displaying the levels of separation between those variables. Intelligence permits insightful action and policy, which obtain better results. The system processes are represented in Figure 2.

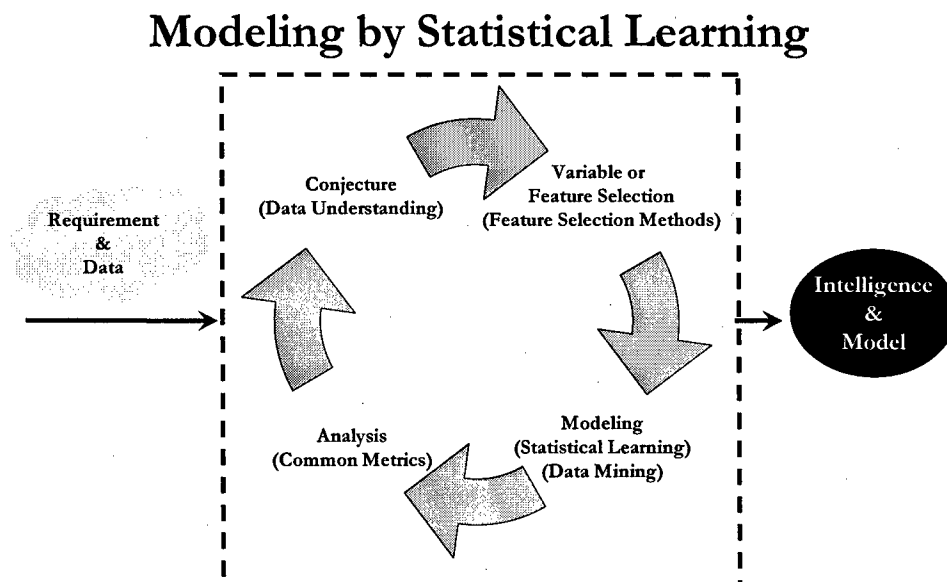


Figure 2, Modeling by Statistical Learning Methods: The Process of Conjecture, Feature Selection, Statistical Learning, and Analysis

The system input is a defined requirement (or even an idea) and the outputs are a model and intelligence. The requirement in this circumstance is bounded by the problem statement and vision. Accompanying the requirement is a collection of data pairs from the NLSI and recruiting production. The primary output is a predictive model that uses all or some of the NLSI data to predict potential production. Four processes sequentially iterate within the system. The four main processes are conjecture, feature selection, statistical learning, and analysis.

### 3.1.2. Conjecture

Initially, we begin with an assertion about the data that specifies which function approximations may accurately predict the response. Multivariate statistics may provide insight into relationships within the data. Preliminary data mining and statistical learning can also provide insight into data relationships and further provide an initial assessment if a function approximation is capable of accurately predicting the data. Either technique or a combination of the techniques provides data understanding.

In subsequent iterations, data understanding is predominately obtained from analysis. Further function approximation assertions are determined by the analysis conducted in the previous cycle. More often, one function approximation dominates other function approximations at the conclusion of the first cycle. Depending on the data and the metric(s) used to evaluate function approximations, it is possible for more than one function approximation in subsequent iterations and occurs when trade-offs exist between function approximations. Common to subsequent iterations is a narrower focus from the previous cycle until little or no gains in predictive accuracy can be obtained.

### 3.1.3. Feature Selection

Feature selection is a very general and powerful method for model performance improvement. Feature selection methods achieve generalization improvements primarily by removing as many noise features as possible. Noise features do not contribute relevant information to the model and potentially cause large variances in prediction. Feature selection methods are not always linear, such as with regression models. They can be non-linear. Greater improvements in feature selection and, subsequently, model performance occur when the feature selection method are constructed within the domain of the function approximation. As an example, if we conjecture the use of logistic regression as a function approximation, then a feature selection method could be a step-wise or a leaps and bounds approach.

Feature selection methods are popular research topics because more powerful methods are domain specific. No matter the function approximation domain, feature selection is mathematically defined as reducing the input space. Without feature selection, the input features are mapped into a new feature space  $\mathbf{x} = (x_1, \dots, x_n) \rightarrow \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_n(\mathbf{x}))$  relevant to the function approximation, where  $n$  is the dimension of the input space. But with feature selection, a feature reduction occurs by  $\mathbf{x} = (x_1, \dots, x_n) \rightarrow \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_d(\mathbf{x}))$ ,  $d < n$  for the function approximation.

### 3.1.4. Statistical Learning

Statistical learning involves the use of more than one function approximation with a goal of discovering a useful function approximation  $\hat{f}(x)$  to the real function  $f(x)$  that underlies the predictive relationship between the features and the response. We may have applied some statistical learning via data mining within the conjecture phase. This process combines the data understanding obtained from the conjecture with feature selection to produce a more refined model.

Dependent upon the response, numerous function approximations exist. Hastie provides a comprehensive catalog of useful function approximations for both classification and regression (Hastie, 2001). Since our application involves a quantitative response, we are interested in regression functions. Although not exhaustive, powerful regression functions can be obtained with linear regression, regression trees, Random Forest, support vector regression, and artificial neural networks. The basic mathematics for each is provided. Detailed expressions can be found in Breiman (Breiman, 1999), Hastie (Hastie, 2001) and Cristianini (Cristianini, 2003).

#### 3.1.4.1 Linear Regression

Linear and multiple regressions are one of the most preferred methods used in prediction (Neter, 1996). They are used frequently throughout science, economics, engineering, business administration, and the social, health, and biological sciences. Recently, linear regression is one of the principle analytical techniques used in Six-Sigma and Lean Six-Sigma (George, et al. 2005). Because of their widespread use, they are understood by many and, consequently, serve as a benchmark for comparing other function approximation predictions. Additionally, linear

regression models have many associated feature selection methods, which also contributes to their popularity.

Many mathematical forms of regression exist. We provide a matrix version due to multiple features and the appeal of linear algebra. The response estimate is provided by the fitted values expressed as:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}. \quad (3.1)$$

Where the vector of fitted values is  $\hat{\mathbf{Y}}$ ,  $\mathbf{X}$  is the data matrix with the first column as ones, and  $\hat{\boldsymbol{\beta}}$  is the estimated coefficients. We define the estimated coefficients as:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}. \quad (3.2)$$

Where the vector of responses is  $\mathbf{Y}$ .

### 3.1.4.2 Tree Regression

Tree based methods are conceptually simple and provide powerful predictors. Tree based methods partition the features into a set of surfaces and then fit a simple model, usually a constant (constants produce flat surfaces), to that surface. Please reference the figure extracted from Hastie's *The Elements of Statistical Learning* to assist with the mathematical explanation that follows.

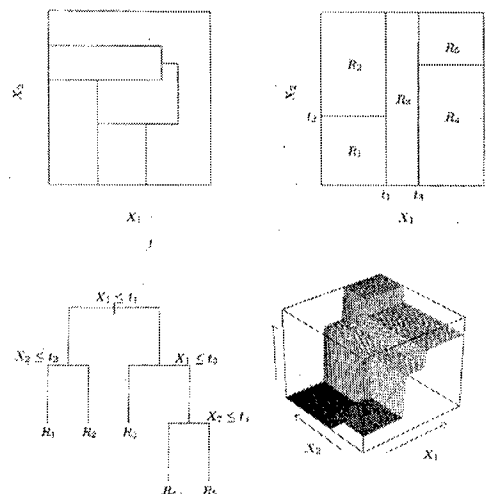


FIGURE 9.2. Partitions and CART. Top right panel shows a partition of a two-dimensional feature space by recursive binary splitting, as used in CART applied to some fake data. Top left panel shows a general partition that cannot be obtained from recursive binary splitting. Bottom left panel shows the tree corresponding to the partition in the top right panel, and a perspective plot of the prediction surface appears in the bottom right panel.

(Hastie, 2001)

Trees are technically grown. The data consists of  $p$  features and a response, for each of the  $N$  observations. A tree algorithm automatically decides the splitting features, the split points of those features, and the topology (the shape) of the tree. Given that we partition into  $M$  regions  $R_1, R_2, \dots, R_M$  and model the response as a constant  $c_m$  in each region, the function is defined by:

$$f(x) = \sum_{m=1}^M c_m I(x \in R_m). \quad (3.3)$$

Typically the criterion minimization is the sum of squared errors,  $\sum (y_i - f(x_i))^2$ , which leaves the best  $\hat{c}_m$  as the average of the response  $y_i$  in region  $R_m$ :

$$\hat{c}_m = \text{avg}(y_i | x_i \in R_m). \quad (3.4)$$

Partitioning is conducted on a single feature and involves a binary split at a split point. Partitioning involves the selection of feature  $j$  and split point  $s$  of the feature. Partitioning is accomplished with a greedy algorithm. We first define the pair of half planes as

$$R_1(j, s) = \{X | X_j \leq s\} \text{ and } R_2(j, s) = \{X | X_j > s\}. \quad (3.5)$$

Seeking the splitting feature and the split point, we solve:

$$\min_{j,s} \left[ \min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]. \quad (3.6)$$

Then for any choice of  $j$  and  $s$ , the inner minimization is solved by applying formula 3.4; i.e.:

$$\hat{c}_1 = \text{avg}(y_i | x_i \in R_1(j, s)) \text{ and } \hat{c}_2 = \text{avg}(y_i | x_i \in R_2(j, s)). \quad (3.7)$$

For each splitting feature, the split point is computed rapidly by scanning all the inputs. Once the best split is found, the data is partitioned into two regions. The process continues to partition regions until a stopping strategy is reached, mostly accomplished by specifying a node size. Otherwise, the splitting could indefinitely continue. Additionally, methods exist for determining optimal trees by applying cost functions to complete or pruned trees. More information is available in Hastie (Hastie, 2001) or Breiman (Breiman, et al., 1998).

Feature selection is inherent to tree regression. The features chosen for splitting were important to the function approximation prediction. However, a tree can only provide a list of features without placing an order that would rank the features from most to less important.

Random Forest, discussed below, is able to order features due to large number of trees and bagging or bootstrapping.

### **3.1.4.3 Random Forest**

Random Forest for regression consists of a random collection of tree based regressions. The function approximation performs nonlinear regression. Randomness is achieved by bagging or bootstrapping the training data set for each grown tree. Because of the Strong Law of Large Numbers, the Random Forest always converges and doesn't over fit the data. As a consequence, through out this research, we grew 500 trees within our Random Forest. Each tree predicts the fitted response and the collection of fitted responses are averaged, yielding a predicted value. Random Forest methods are proven to significantly increase tree based function prediction accuracy (Brieman, 1999).

### **3.1.4.4 Support Vector Regression**

Support vector machines and regression are recent developments in function approximation. They are able to predict in nonlinear and high feature space environments without over fit. As a result, they recently have had tremendous success in social, health, and biological sciences. Uniquely, support vector regression defines a prediction boundary based on observations rather than features. A benefit of predicting by observations rather than features is the ability to predict anomalies, as well as the majority, within the data. A major concern with support vector regression is a requirement for a large number of training observations in order to explain most future unseen observations. Smaller training data sets produce larger errors (Halstead, 2005), (Hastie, 2001).

Because support vector regression predicts with observations (support vectors), feature selection methods are prone to including noise features (Halstead, 2005), (Hastie, 2001), (Cristianini, 2000), (Scholkopf, 2002). Noise feature inclusion, as with support vector function approximation, is reduced with larger training data sizes. Although it is preferable to use feature selection methods germane to the function approximation, features provided by other methods can be suitable for support vector regression.

The theoretical development of SVR is similar to those of the SVM. In SVM, the slack and the norm of the weight vector bind the classifier. Within SVR, bounding is achieved by the

loss function, which is equivalent to the slack. The loss function ignores errors within a defined distance from the true value. For SVR,  $\varepsilon$ -insensitive represents the loss instead of the generalization error. By using  $\varepsilon$ -insensitive, SVR achieves sparseness.

The quadratic  $\varepsilon$ -insensitive loss function is formally defined as:

$$L_2^\varepsilon(\mathbf{x}, y, f) = |y - f(\mathbf{x})|_\varepsilon^2. \quad (3.8)$$

The  $\varepsilon$ -insensitive loss function can be used to develop quadratic, linear, and ridge regression SVR models. As an example, we highlight the quadratic  $\varepsilon$ -insensitive loss SVR. Minimizing the sum of the quadratic  $\varepsilon$ -insensitive losses optimizes the generalization of a regression:

$$R^2 \|\mathbf{w}\|^2 + \sum_{i=1}^l L_2^\varepsilon(\mathbf{x}_i, y_i, f), \quad (3.9)$$

where  $f$  is a function defined by the weight vector  $\mathbf{w}$  and  $R$  is the radius of the data ball. By minimizing equation (3.9), all bounds of the geometric margins  $\gamma$  are also minimized. The parameter  $C$  measures the trade-off between complexity and losses. The primal problem is:

$$\begin{aligned} \text{Min} \quad & \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i^2 + \hat{\xi}_i^2), \\ \text{ST} \quad & (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - y_i \leq \varepsilon + \xi_i, \quad i = 1, \dots, l, \\ & y_i - (\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \leq \varepsilon + \hat{\xi}_i, \quad i = 1, \dots, l, \\ & \xi_i, \hat{\xi}_i \geq 0, \quad i = 1, \dots, l \end{aligned} \quad (3.10)$$

where the two new slack variables are used for exceeding or being below the target value of  $\varepsilon$ .

The consequent Lagrange multipliers are:

$$\begin{aligned} \text{Max:} \quad & \sum_{i=1}^l y_i (\bar{\alpha}_i - \alpha_i) - \varepsilon \sum_{i=1}^l (\bar{\alpha}_i + \alpha_i) \\ & - \frac{1}{2} \sum_{i,j=1}^l (\bar{\alpha}_i - \alpha_i) (\bar{\alpha}_j - \alpha_j) \left( \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + \frac{1}{C} \delta_{ij} \right), \\ \text{ST:} \quad & \sum_{i=1}^l (\bar{\alpha}_i + \alpha_i) = 0 \\ & \alpha_i, \bar{\alpha}_i \geq 0, \quad i = 1, \dots, l \end{aligned} \quad (3.11)$$

By substituting  $\beta = \bar{\alpha} - \alpha$  and using the relation  $\alpha_i \bar{\alpha}_i = 0$ , we rewrite the dual to closely resemble a classification case:



$$\begin{aligned}
Max: & \sum_{i=1}^l y_i (\beta_i) - \varepsilon \sum_{i=1}^l |\beta_i| - \frac{1}{2} \sum_{i,j=1}^l \beta_i \beta_j \left( \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \frac{1}{C} \delta_{ij} \right) \\
ST: & \sum_{i=1}^l \beta_i = 0, \quad i = 1, \dots, l
\end{aligned} \tag{3.12}$$

When the  $\varepsilon = 0$  in the  $y_i \in \{-1, 1\}$ , the similarity becomes apparent with the use of variables  $\bar{\beta}_i = y_i \beta_i$ . The difference is  $\bar{\beta}_i$  is not constrained to be positive unlike the corresponding  $\alpha_i$  in the classification case.

For non-zero  $\varepsilon$  an extra weight decay parameter  $C$  is added to approximate standard least squares linear regression. As  $C \rightarrow \infty$ , the SVR approximates unconstrained least squared and leaves the inner product matrix diagonal unchanged. This result leads to a more general kernel regression version.

Given a training set of data  $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l))$  and the implicit feature space defined by kernel  $K(\mathbf{x}, \mathbf{z})$ , assume  $\alpha_i^*$  solves the quadratic:

$$\begin{aligned}
Max: & \sum_{i=1}^l y_i \alpha_i - \varepsilon \sum_{i=1}^l |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j \left( K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{C} \delta_{ij} \right) \\
ST: & \sum_{i=1}^l \alpha_i = 0, \quad i = 1, \dots, l
\end{aligned} \tag{3.13}$$

We let  $f(\mathbf{x}) = \sum_{i=1}^l \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$ , where  $b^*$  is chosen to force  $f(\mathbf{x}_i) - y_i = -\varepsilon - \alpha_i^*/C$  for any  $i$  with  $\alpha_i^* > 0$ . This makes the function  $f(\mathbf{x})$  equivalent to the hyper plane in feature space defined by the kernel  $K(\mathbf{x}, \mathbf{x})$ .

### 3.1.5. Analysis

Comparing different function approximations can be challenging. For example, volumes of text, research, and professional journal articles address linear regression metrics. As a consequence, there are accepted industry standards for evaluating the quality of a linear regression model. Yet the metrics associated with other function approximations are not necessarily equivalent potentially resulting in a comparison of different metrics.

The analysis must, therefore, compare function approximations with a common metric. Because we are interested in predictive accuracy, common accuracy measurements should be explored. The accuracy measurements are performed on feature and response data pairs that are withheld from the training data. These data are test and/or validation data, constituting a new

unobserved data that happens to have a recorded response. Typically, errors are used to determine accuracy. For classification, the computation of total error, type I error, and type II error are straightforward and involve classical methods. For regression, many forms of sum or square errors may be used.

For this application, we developed a type of mean square error that did not account for model parameter degrees of freedom. This is acceptable since we do not wish to penalize a function approximation for its parameters under these conditions. We used the form:

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (3.14)$$

In subsequent analysis during further iterations, other statistical methods become available. The purpose of further analysis may change from comparison to sensitivity. As such, these subsequent methods may include bootstrapping and the resulting traditional statistics provided by the bootstrapped collection of performance metrics.

### 3.1.6. Model Refinement

Model refinements occur in subsequent iterations of the processes. We focus our efforts in feature selection and the resulting function approximation. The purpose remains prediction accuracy. We also include sensitivity to ensure model stability.

Refining feature selection methods depends on the function approximation. For linear regression, the initial feature selection will not change. However, for methods such as support vector regression or Random Forest, possibilities exist for improving the feature set. Applying an error metric through a greedy algorithm or a branch and bound algorithm can refine the feature selection and improve model generalization. Bootstrapping the model's validation data provides insight into the model's sensitivity. Bootstrapping is withholding random observations with replacement from the validation data in order to obtain multiple metric measurements, which enables traditional statistical methods. Bootstrapping obtains a mean error metric and standard deviation, this facilitates other statistical tests.

## 3.2 Data

### 3.2.1. Response Variable

The response variable is a metric representing recruiter production, gross writer rate. Gross write rate measures the average number of contracts obtained by a recruiter in one month.

The first six months of production are removed from the average to account for a learning curve (PDRI, 2005).

### **3.2.2. Prediction Variables (Features)**

The data contained 260 features, most representing the NLSI test question bank and others representing biographical data, such as age and gender. There were a total of 1,954 observations prior to data cleansing. A complete listing of feature definitions is contained in Appendix B.

#### **3.2.2.1 Data Cleansing**

The NLSI part II questions required data manipulations due to its structure. Each question in part II generates four features, each feature represents an answer. The questions required binary coding. If the question is answered, then it receives a value of one. Otherwise, unanswered questions receive a zero.

The data were cleaned to remove blanks within a feature within the NLSI part I. If a feature had a blank, the entire observation was removed from the data. Blanks in the data most likely occurred from test takers not answering questions. Because of blanks, 41 observations were removed. We had 1,913 observations for training and validation.

#### **3.2.2.2 Training and Validation Data**

The training and validation data were randomly removed without placement from the data containing 1,913 observations. The data were then randomized within their training and validation data sets. We created two training data sets and two validation data sets. One training and validation data set pair was used for Random Forest and SVM 2. Those data contained 1,000 training observations and 676 validation observations. The other pair was used for the linear regression and SVM 1 and contained 888 training observations and 610 validation observations.

## **Chapter 4:      Modeling Methodology Results**

### ***4.1 Initial Data Understanding***

We used a combination of multivariate statistics and data mining within Clementine 9.0 and R statistical language. The multivariate statistics used were the variance-covariance tables, the multivariate correlation matrix, and a pair-wise scatter plot of the NLSI. The initial data mining techniques used tree regression, linear regressions, and artificial neural networks to discover which function approximations may be more promising than others. Clementine lacks support vector capability.

From the data understanding, we noticed NLSI score and age were significant features for predicting gross write rate. We also discovered trees and linear regression would substantially outperform artificial neural networks. This information guides the conjecture to explore Random Forest, linear regression, and support vector regression. Random Forest inherently conducts a form of feature selection. We conducted linear regression feature selection with step-wise approach in both directions. We also used the features extracted from Random Forest and step-wise regression in two support vector regressions, each using the features from one of the two feature selection methods.

### ***4.2 Initial Feature Selection***

Table 1 and Table 2 contain the features selected by step-wise regression and Random Forest, respectively. Both tables are ordered from the most important feature used by the function approximation to the better features used by the function approximation. It would be premature to use the tables for asserting which features are relevant and important to predicting gross write rate. For this reason, we do not provide a comparison between the two feature lists. Rather, these features demonstrate the input vectors for the four models explored in the initial statistical learning. As a set, they contribute information that enables each function approximation to predict potential gross write rate. Further model analysis can lead to assertions of individual feature selection.

**Table 1, Ordered Step-wise Feature Selection Results (Ordered from Best to Better Features)**

<b>Feature</b>	<b>Df</b>	<b>Sum Sq</b>	<b>Mean Sq</b>	<b>F Value</b>	<b>Pr(&gt;F)</b>
BIO022	1	5.189	5.189	19.6582	1.05E-05
BIO004	1	4.936	4.936	18.7004	1.71E-05
BIO046	1	2.732	2.732	10.3489	0.001344
LEAD32	1	2.455	2.455	9.2997	0.002362
AGE	1	1.885	1.885	7.14	0.00768
BIO025	1	1.876	1.876	7.1092	0.007812
BIO057	1	1.791	1.791	6.7837	0.009357
BIO106	1	1.237	1.237	4.6863	0.030675
LIE19	1	1.187	1.187	4.4983	0.034212
LEAD27	1	1.174	1.174	4.4487	0.035214
LEAD15	1	1.151	1.151	4.3598	0.037088
BIO049	1	1.066	1.066	4.0384	0.044786
BIO079	1	1.001	1.001	3.7909	0.051854
BIO078	1	0.893	0.893	3.3835	0.066192
ADJ03	1	0.729	0.729	2.7609	0.096957
BIO071	1	0.727	0.727	2.7545	0.097345
BIO023	1	0.524	0.524	1.9851	0.159216
ADJ15	1	0.328	0.328	1.2425	0.265292
PC32	1	0.317	0.317	1.2	0.273623
BIO093	1	0.141	0.141	0.5348	0.464774

**Table 2, Ordered Random Forest Feature Selection Results (Ordered from Best to Better)**

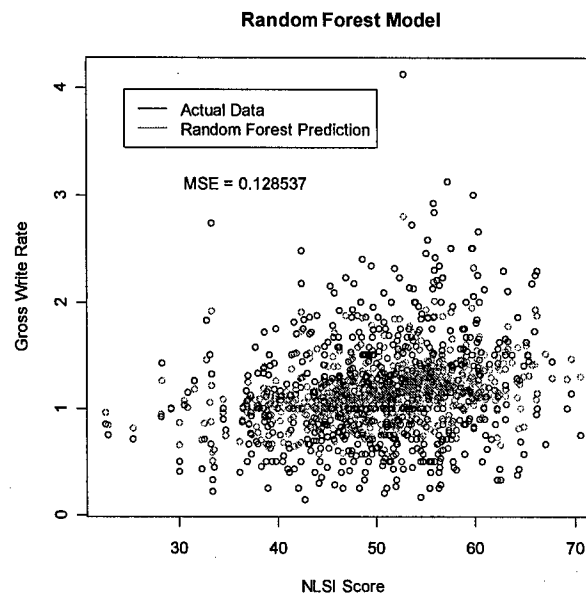
<b>Feature</b>	<b>ImportanceScore</b>
AGE	20.68
NLSIPVSC	19.48
BIO109	7.69
BIO107	6.55
BIO046	5.61
BIO018	5.37
BIO023	5.22
BIO095	5.16
BIO113	5
BIO049	4.99
BIO007	4.67
BIO115	4.67
BIO041	4.62
BIO102	4.56
BIO006	4.55
BIO038	4.53
BIO037	4.5
BIO087	4.39
BIO054	4.37
BIO066	4.36
BIO001	4.32
BIO008	4.28
BIO064	4.27
BIO050	4.24
BIO078	4.22
BIO071	4.19
BIO040	4.18
BIO116	4.16
BIO014	4
BIO103	3.95
BIO090	3.92
BIO110	3.88
BIO061	3.83
BIO033	3.76
BIO088	3.76
BIO093	3.76
BIO022	3.7
BIO082	3.55
BIO039	3.52
BIO118	3.36
BIO069	3.33
ADJ32	3.28
BIO017	3.24
BIO057	3.16
BIO075	3.14
BIO108	2.99
BIO053	2.94
BIO106	2.94
DEPEND22	2.79
BIO016	2.78
BIO028	2.76
DEP17	2.71
SEX	2.65
BIO051	2.6
LEAD12	2.49
AGREE23	2.36
BIO094	2.2
BIO070	2.14

### **4.3 Initial Statistical Learning**

We conducted four statistical learning algorithms: linear regression, Random Forest, and two support vector regression. The features obtained from step-wise regression were used as the

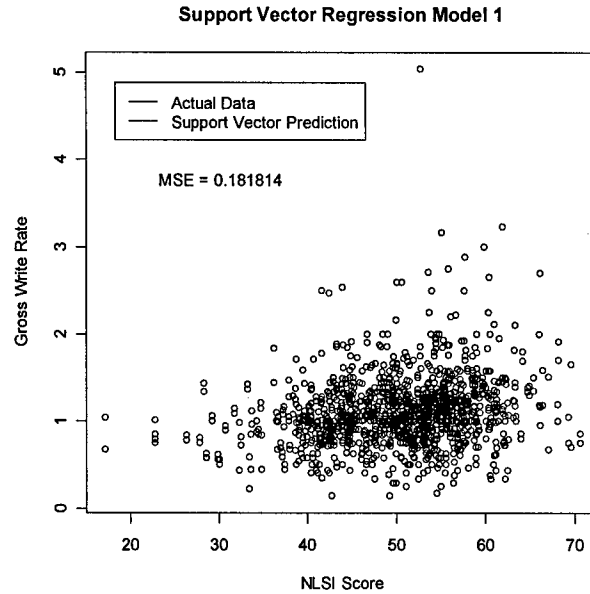
input vector (data pairs) in the linear regression model and the first support vector regression model (SVR 1). While the features obtained from Random Forest were used as the input vector for Random Forest and the second support vector regression model (SVR 2).

To visualize each of the models, we plotted the estimated GWR along with the true GWR using the NLSI score as an index feature. Validation data were used in the plot and prediction. The following model plots are sequentially ordered from best to worst. The order from best to worst is Random Forest, SVR 1, SVR 2, and linear regression.



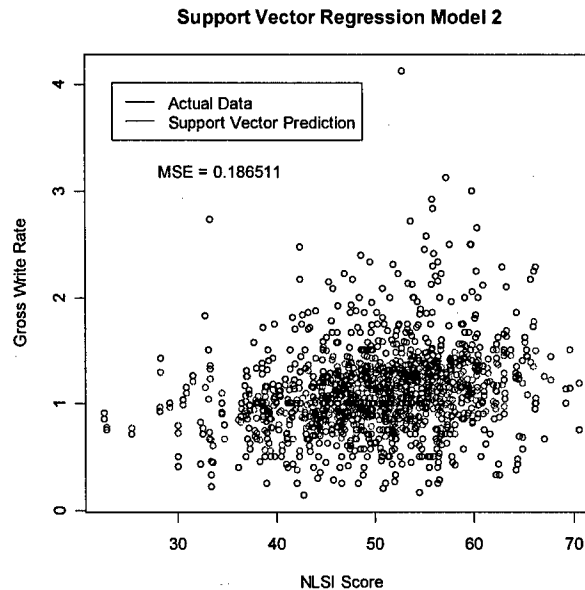
**Figure 3, Random Forest Model, Predicted Values Are Green and Actual Observations Are Black**

Random Forest was the best model not only for its lower mean square error. It better predicted the majority of the observations and was able to better predict anomalies within the data. This is most evident by comparing Random Forest (Figure 3) with linear regression (Figure 6). The Random Forest model produces a positively related and curved prediction of GWR for increases in NLSI score. The linear regression model produces a less positively related and linear prediction of GWR. The Random Forest model reaches out and better predicts anomalies, while the linear regression model is more conservative towards anomalies, fitting them with the majority.



**Figure 4, Support Vector Regression Model 1 (SVR 1), Predicted Values Are Blue and Actual Observations Are Black**

Both support vector regressions performed better than linear regression. Both show some predicted value curvature in GWR as the NLSI score increases. Both are less restrictive with anomalies than linear regression, yet are not as flexible towards satisfying anomalies as Random Forest. They also were not as adept as Random Forest when explaining the majority.



**Figure 5, Support Vector Regression Model 2 (SVR 2), Predicted Values Are Red and Actual Observations Are Black**



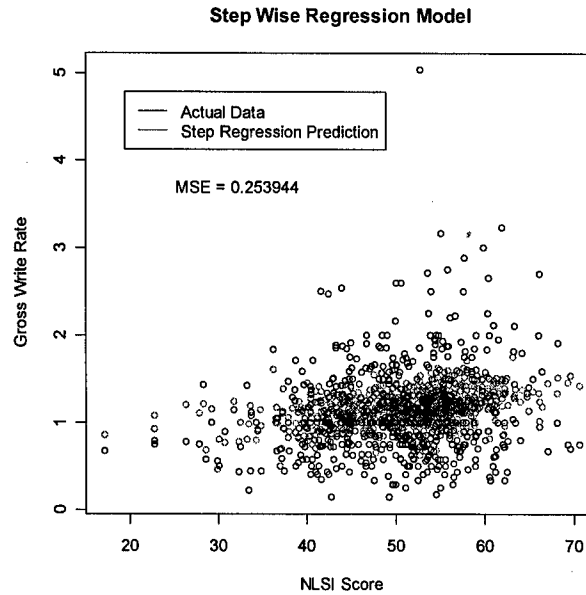


Figure 6, Linear Regression Model, Predicted Values Are Purple and Actual Observations Are Black

The linear regression model was not a good predictor of GWR with the data. The model plot demonstrates a positive relationship between predicting GWR and NLSI score, but the model lacks the fidelity required to explain the data.

#### 4.4 Initial Analysis

Our initial decision criterion is MSE (Equation 3.14). The best model has the lower MSE, demonstrating better model prediction by a lower error rate. Random Forest dominates the other models in accuracy (Table 3).

Table 3, Model MSE Comparison

Model	MSE
Random Forrest	0.128537
SVR 1	0.181814
SVR 2	0.186511
Linear Regression	0.253944

Because Random Forest is significantly more accurate than the other function approximations for the data, especially linear regression, we continue to only refine the Random Forest function approximation and features to produce the best prediction model for the data.

The Random Forest model quickly converged to the lower error rate by the time it grew one hundred trees. Because Random Forest is not subjected to over fit, we continue to grow 500 trees in each subsequent Random Forest model. Figure 7 demonstrates a rapid decrease in error related to forest growth and stabilization of that error past one hundred trees.

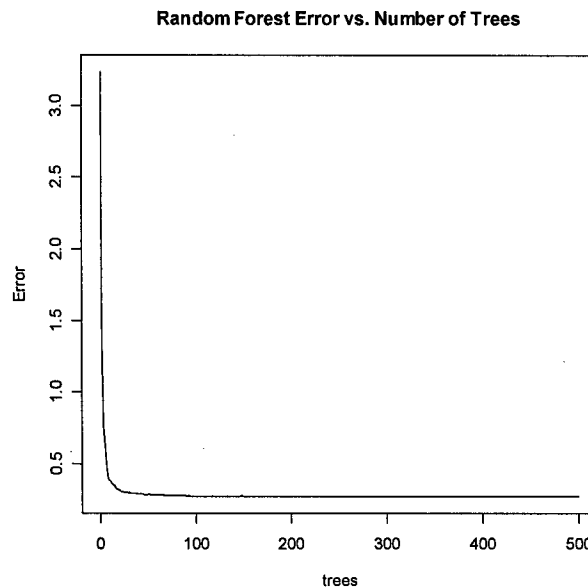


Figure 7, Random Forest Error Rates as Related to the Number of Trees Grown within the Forest

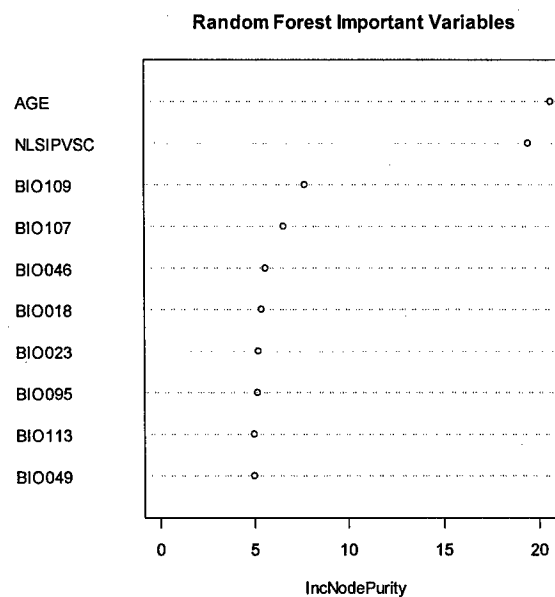


Figure 8, Top 10 Random Forest Features

Figure 8 is a plot of the top ten Random Forest features. By the separation achieved by age and NLSI score, we know that many trees split the response based on various split points for age and NLSI score. Because of the data complexity, there is no right answer to the correct mix of age and NLSI score. Earlier data mining in Clementine strongly suggested multi-modal peaks for various combinations of age and NLSI score in the data. Unfortunately, this doesn't assist policy making to determine the best ages for recruiting since there is no clear cut answer. However, we do know that age and NLSI score are not related to each other. Figure 9 demonstrates a lack of correlation between age and NLSI score by the random scatter of its two plots. This indicates NLSI score is resilient to age. As a result, NLSI scores for an individual soldier should remain constant no matter the soldier's age. Administration of the NLSI score at any age should produce a constant result for any individual.

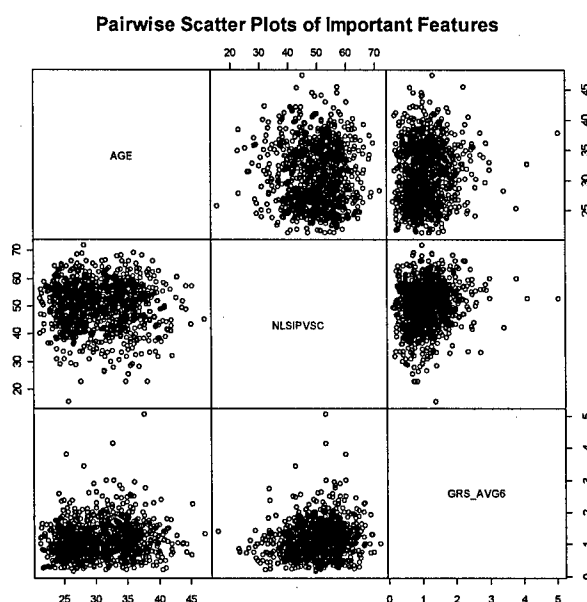
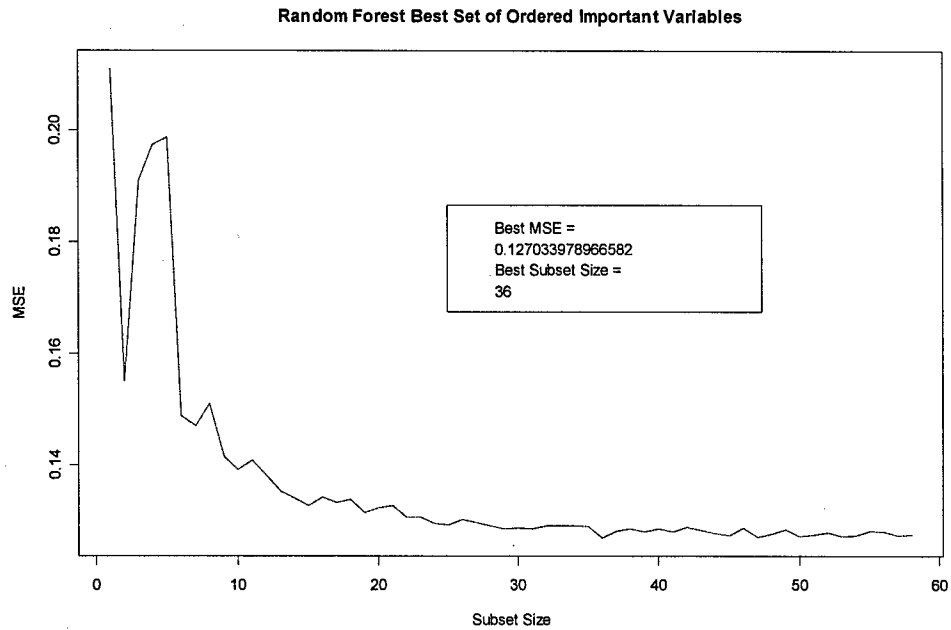


Figure 9, Pairwise Scatter Plots of Age, NLSI Score, and Gross Write Rate

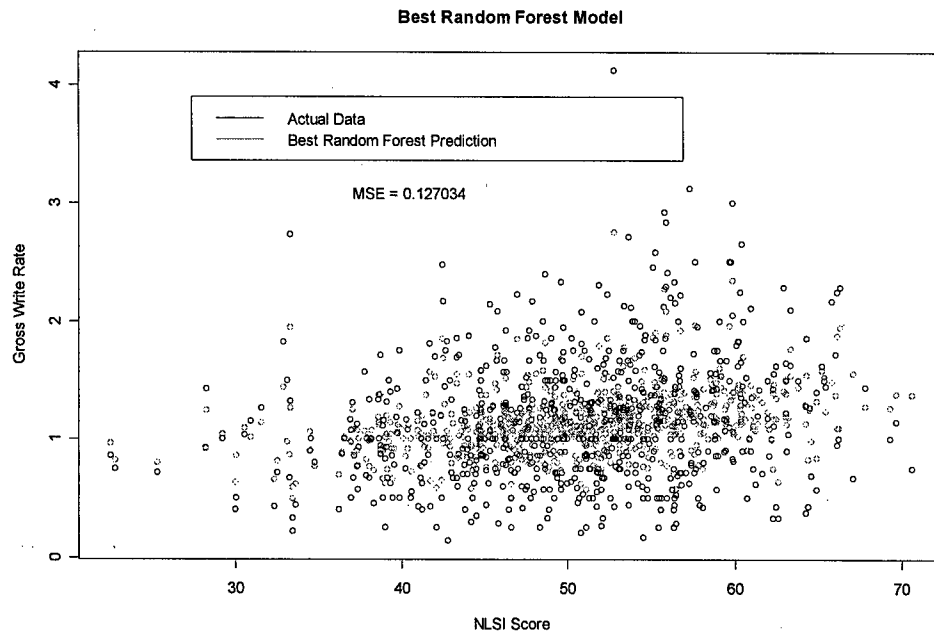
#### 4.5 Refinements

Refinements focus on improving Random Forest model accuracy by improving feature selection. Feature selection is improved by conducting a greedy algorithm that searches for the best feature subset size from a set of order features using minimum mean square error as the criterion. Later when TAIS data is available, feature selection will include the TAIS data and determine the best feature set using the same greedy algorithm. Figure 10 demonstrates the results of the greedy algorithm search for the best subset from an ordered feature set. The best

subset for Random Forest on these data occurs with the first thirty six individually good features. We reduce the model's mean square error to 0.1270 from the previous value of 0.1285 (Figure 11). Those features prior to and including the 36 features provide the Random Forest model relevant information. Those features after these contribute noise and will not improve the model. The noise features, however, can cause model instability. Therefore, we remove them from the model.

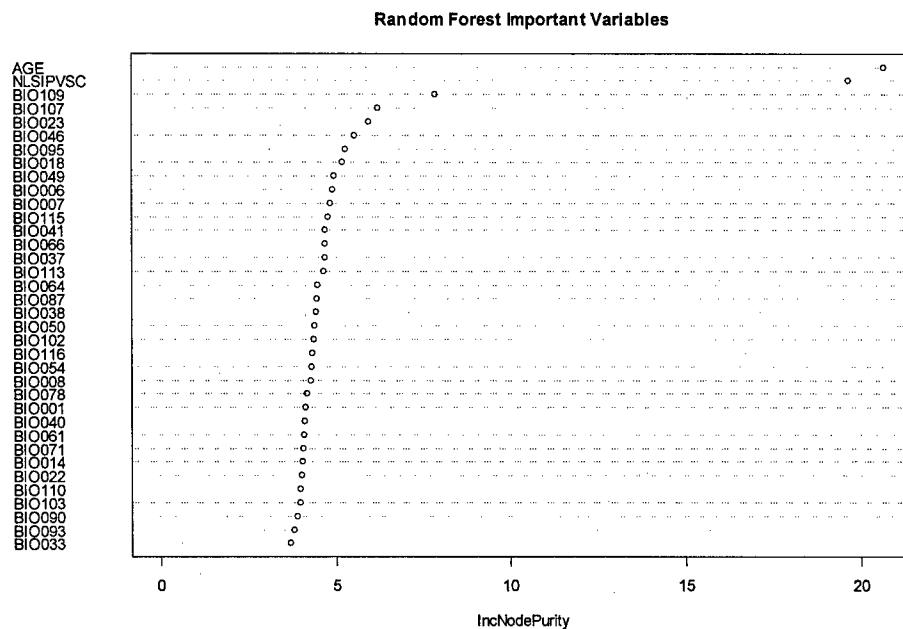


**Figure 10, Random Forest Best Subset Selection as Determined by Minimum MSE**



**Figure 11, Improved Random Forest Model via Feature Selection from an Ordered Greedy Algorithm**

The best 36 Random Forest features are depicted in Figure 12. Age and NLSI score remain the most influential features and achieve significant separation from the other good features. The BIO109 NLSI question also has some separation from the other important features. These features become the input vector for all further Random Forest models.



**Figure 12, Random Forest Important Model Features**

## Chapter 5: Model

### 5.1 Model Specifications

#### 5.1.1. Prediction Model

The Recruiter Selection Model is an eloquent prediction model that predicts potential gross write rate for an individual, creates an order of merit list based on potential gross write rate, and provides an order of merit cumulative gross write estimate. The model demonstrates innovation by transforming a complex theoretical application to an operationally simple model. The function approximation model is coded in R statistical language and uses Excel and VBA to call and control the R program. Excel and VBA enable the use of this sophisticated model, which would typically be reserved for advanced engineers, economists, and statisticians. The final model is a push-button application within Excel.

The Recruiter Selection Model is a statistically sophisticated model. Because of the sophistication, it is stringent in data structures and software requirements, without much flexibility in either. An operator must load data and install the model precisely as described to ensure operation.

Much care must be given to the data. The recruiter selection model uses raw NLSI and age data provided by PDRI. The data must have the structure explained in paragraph 5.2.1 Model Prediction Data. The data must be saved in a tab delimited text file called *nlsi.txt*, which is case sensitive. The data variable headers are in caps and must be labeled and ordered as explained in paragraph 5.2.1. The data can't contain blank spaces and observations must be numeric, except for social security numbers. Excel provides a convenient method for saving flat text files in this method. The text file must be saved in a folder created and labeled *c:/data*. An example data set is provided.

The program requires two software installations and some package installations from R. The programs are R and RExcel and the packages are randomForest, MASS, tree, lattice, graphics, car, RColorBrewer, and class. The Recruiter Selection Model assists with the download by providing the required sites and relevant information. When you open *RecruiterSelect(nlsi).xls*, the United States Army Recruiter Selection Model window appears (Figure 13). The window contains three buttons, which are the icons: information, R program, and recruiter badge. Basic technical information is contained in the information button. The

software installation is contained in the R program button. The prediction model is contained within the recruiter badge button.

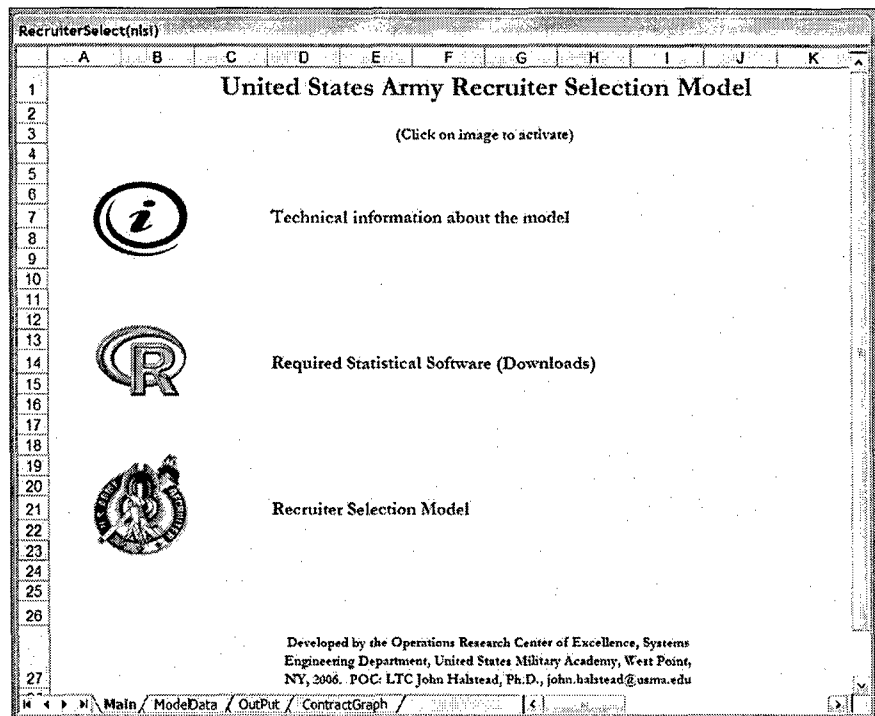


Figure 13, U.S. Army Recruiter Selection Model Window

The information button contains technical information that summarizes the development of the model and the software installation. The About the Model window is displayed in Figure 14.

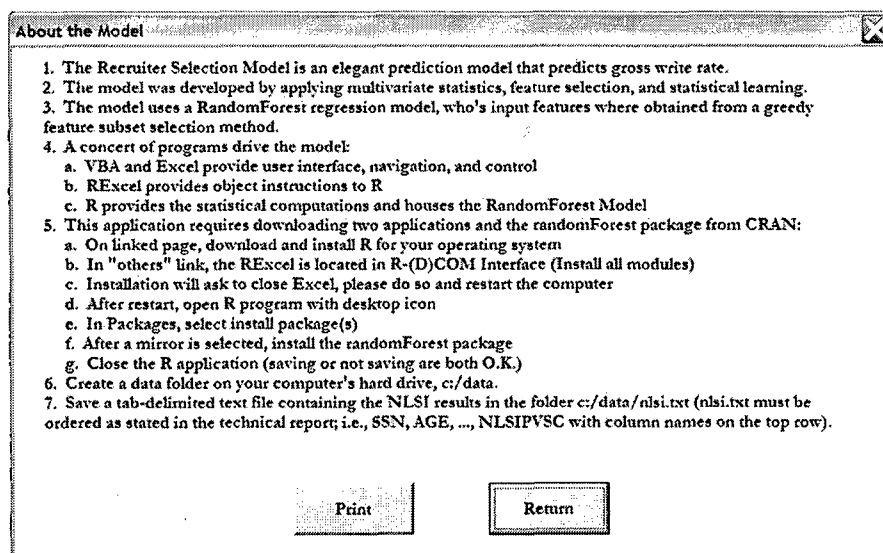


Figure 14, About the Model Window

The R program button provides a link to the software site, CRAN, which contains the two required programs. One required program is R and the other is RExcel. The R program provides the sophisticated statistical model, while RExcel provides an Excel add-on that assists VBA with calling and running R programs. Download the R program first and install it. The download and install RExcel. Installing RExcel will require you to close the Recruiter Selection Model and, after installation, re-start your computer. After R and RExcel have been installed, the randomForest package is installed by accessing R. Take the following steps when installing the required software and package.

Step 1: Obtain the R program. Open the R link that takes you to the Comprehensive R Archive Network (CRAN) site and select *Download and Install R for the Windows (95 and later)* (Figure 15).

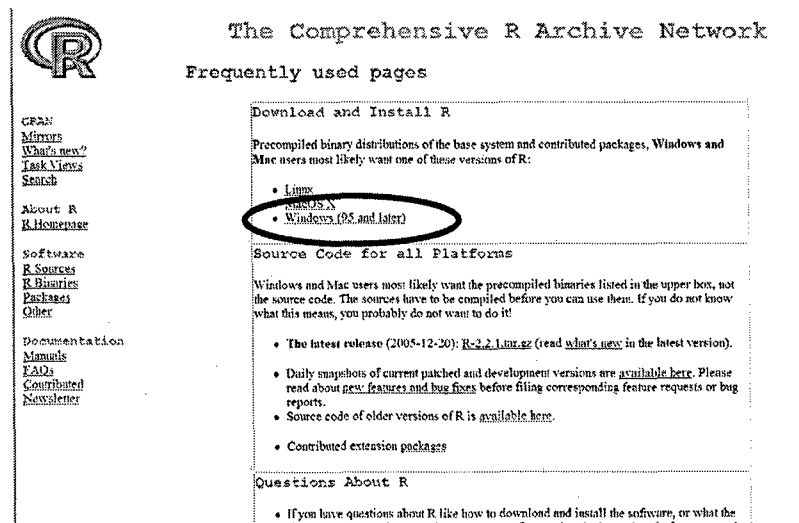


Figure 15, Comprehensive R Archive Network (CRAN) Site

Select the base subdirectory (Figure 16).



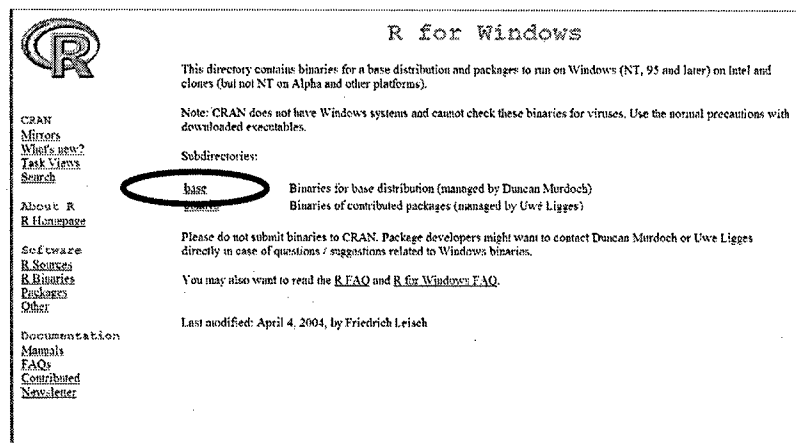


Figure 16, R for Windows Site

Select the *R-2.2.1-win32.exe* link (Figure 17). This link name will change with software updates. Also, this link will activate an installation window (Figure 18). You should select *Run* and then follow all R installation directions.

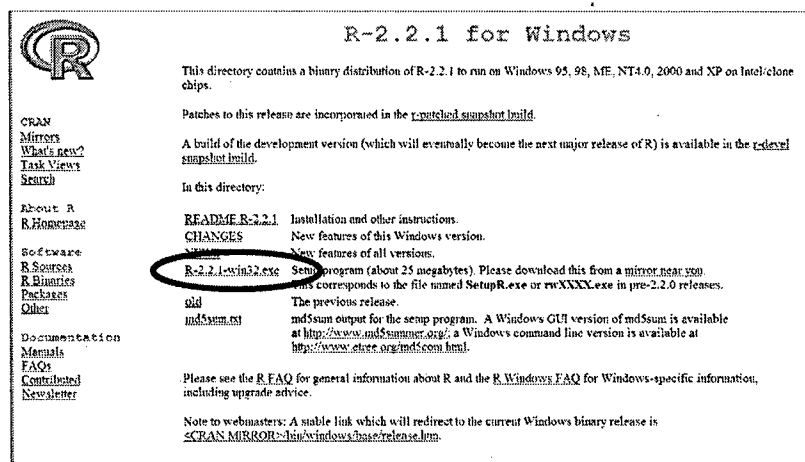


Figure 17, R-2.2.1 for Windows Site

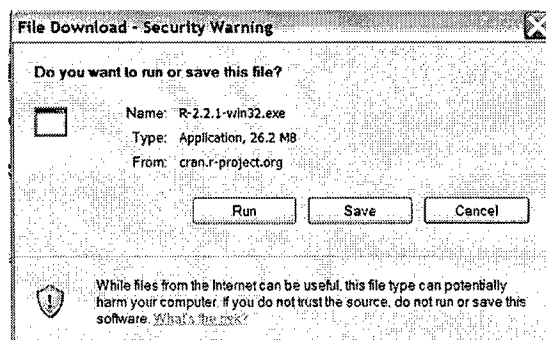
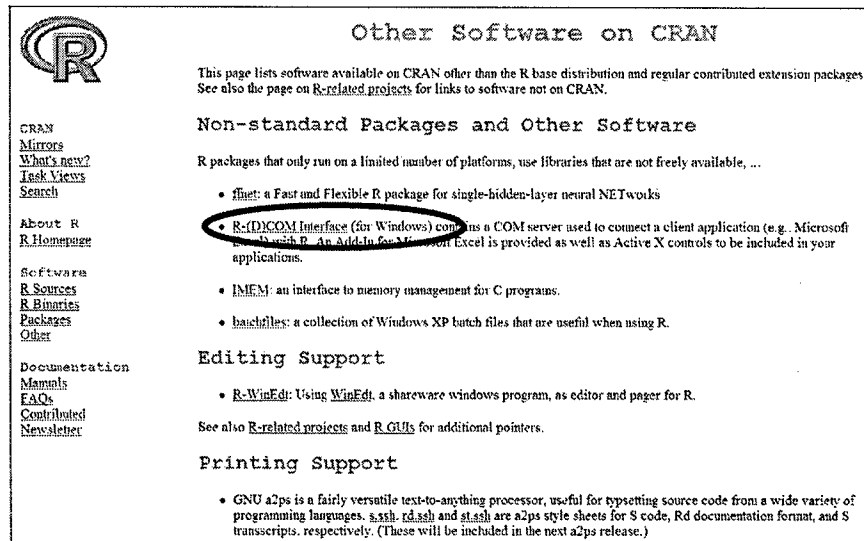


Figure 18, File Download Window

Step 2: After installing R, download RExcel. RExcel is located in the other software on CRAN web page and is access by selecting *Other* (located on the right side bar just below the R icon) on any of the CRAN web pages. Once on the *Other Software on CRAN* web page, select the *R-(D)COM Interface (for Windows)* link (Figure 19).



**Other Software on CRAN**

This page lists software available on CRAN other than the R base distribution and regular contributed extension packages. See also the page on [R-related projects](#) for links to software not on CRAN.

**Non-standard Packages and Other Software**

R packages that only run on a limited number of platforms, use libraries that are not freely available, ...

- [flnet](#): a Fast and Flexible R package for single-hidden-layer neural NETWORKs
- [R-\(D\)COM Interface \(for Windows\)](#) contains a COM server used to connect a client application (e.g. Microsoft Excel) with R. An Add-In for Microsoft Excel is provided as well as ActiveX controls to be included in your applications.
- [IMEM](#): an interface to memory management for C programs.
- [batchfiles](#): a collection of Windows XP batch files that are useful when using R.

**Editing Support**

- [R-WinEdi](#): Using [WinEdi](#), a shareware windows program, as editor and pager for R.

See also [R-related projects](#) and [R GUIs](#) for additional pointers.

**Printing Support**

- [GNU a2ps](#) is a fairly versatile text-to-anything processor, useful for typesetting source code from a wide variety of programming languages. [a2ps](#), [a2ps](#) and [a2ps](#) are a2ps style sheets for S code, Rd documentation format, and S transcripts, respectively. (These will be included in the next a2ps release.)

Figure 19, Other Software on CRAN Site

The RExcel add-on is provided by downloading and installing *RSrv200.3x3* (or a more recent version indicated by a higher number) as shown in Figure 20. Close the Recruiter Selection Model so the Excel add-on can be installed on the computer.



**Index of /contrib/extra/dcom**

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	05-Jul-2005 11:24	-	
<a href="#">RSrv125.exe</a>	22-Jun-2004 19:44	2.7M	
<a href="#">RSrv125.html</a>	22-Jun-2004 19:44	23K	
<a href="#">RSrv125.zip</a>	22-Jun-2004 19:44	192K	
<a href="#">RSrv200.exe</a>	02-Dec-2005 14:30	2.8M	
<a href="#">RSrv200.html</a>	02-Dec-2005 14:31	10K	

Apache/1.3.33 Server at cran.r-project.org Port 80

Figure 20, Index of RExcel Software Site

Selecting *RSrv200.3x3* will initiate another installation window. Again, the *Run* button should be selected. For the best results, select all the offered components as illustrated in Figure 21. After installation, the computer should be restarted. The next time Excel is used, a RExcel menu will be added to the toolbar.

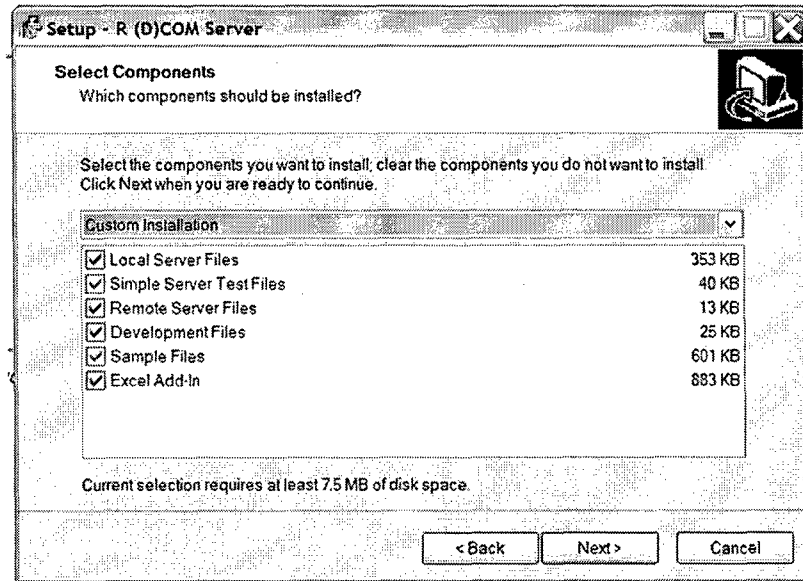


Figure 21, Setup R (D)COM Server Window

Step 3: Once the computer restarts, open R with the new shortcut icon on the desktop. The randomForest package is installation occurs within R. Once R is open select the *Packages* menu and select *Install package(s)*, as demonstrated in Figure 22. The package is downloaded from one of many mirror sites; select a mirror from within the United States (Figure 23).

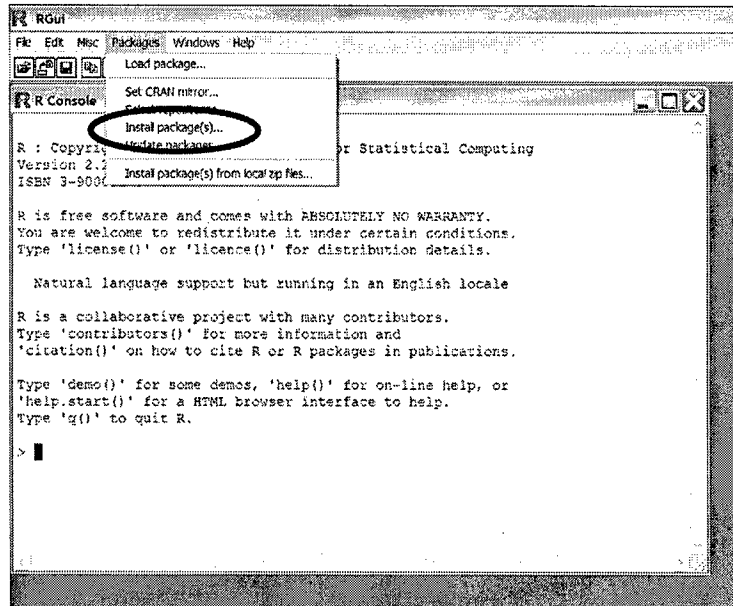


Figure 22, RGui Package Installation Window

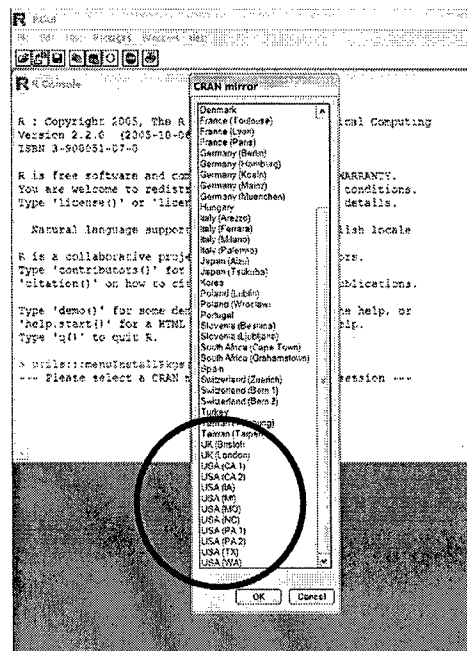


Figure 23, CRAN Mirror Window

Once the *Packages* window appears, scroll to the *randomForest* and select it (Figure 24). The package will automatically install to your R program. Repeat this step for the remaining packages (*MASS*, *tree*, *lattice*, *graphics*, *car*, *RColorBrewer*, and *class*). When ever an R

program is manually upgraded by CRAN, the *randomForest* package must be manually re-installed too.

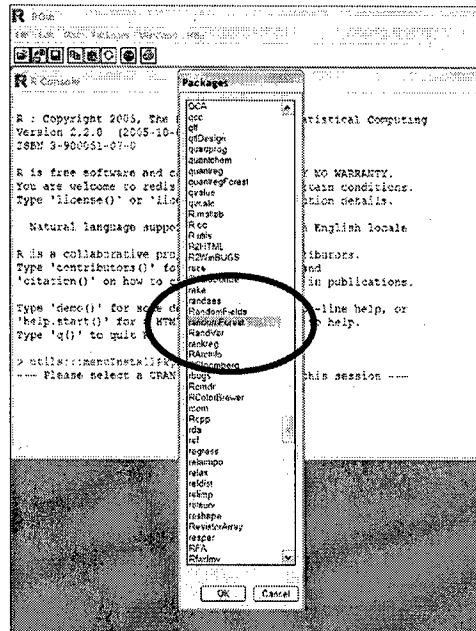


Figure 24, Packages Window

When both software and the package are installed, the Recruiter Selection Model can be run by selecting the recruiter badge. Once the recruiter badge is selected the Recruiter Selection Model window appears as in Figure 25. (Please note, the R program, RExcel, and randomForest package installation occurs only on the initial installation and for upgrades, which should be checked on CRAN every six months.)

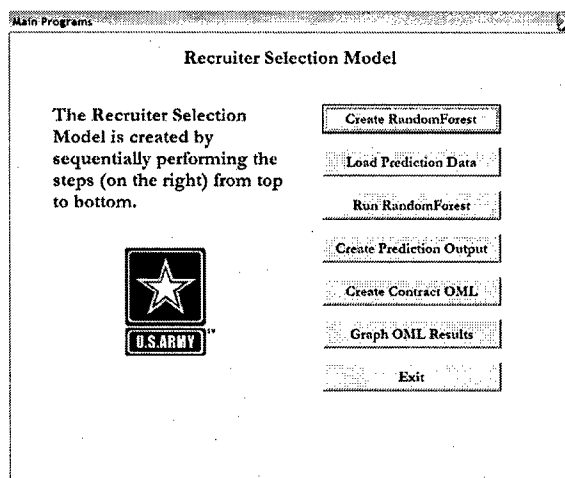
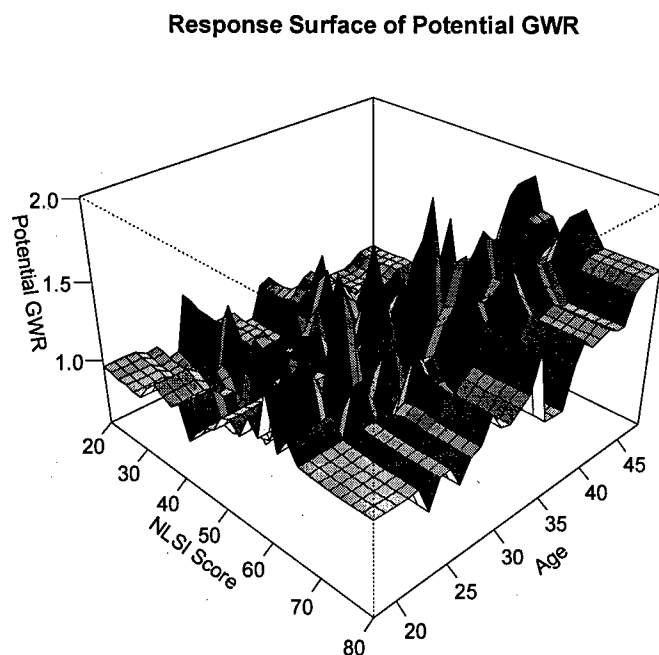


Figure 25, Recruiter Selection Model Window

The model is created by sequentially selecting the buttons on the right side of the window from top to bottom. Message boxes assist with navigation. The model creates a RandomForest regression prediction model when *Create RandomForest* is selected. The *Load Prediction Data* button places the *nlsi.txt* file into the model. The *Run RandomForest* button uses the *nlsi.txt* data to predict the potential gross write rate for each individual contained in the file. The *Create Prediction Output* button calls the predicted gross write rate and the individual social security numbers from the R program and pastes them into the *OutPut* worksheet. It also opens the *OutPut* worksheet. The *Create Contract OML* is an Excel macro that sorts the individual gross write rate from highest to lowest (creating an order of merit list (OML)), calculates an estimated annual gross contract number for each individual, and the cumulative estimated gross write rate for the OML. The *Graph OML Results* button navigates to an output graph that visualizes the OML individual performance and total gross contracts obtained as the OML progresses from highest to lowest GWR. The *Exit* button closes the program and permits copy and paste operations on the OML and the graph. The model data and the front graphic user interface are protected.

### 5.1.2. Recruiter Order of Merit Ranking and Predicted Gross Write Rate

The model is necessary for selecting recruiters. Establishing a NLSI cut off score or a range of NLSI scores for the purpose of selecting recruiters will not accomplish the selection of the right recruiting force and not provide a return on investment. The model uses 36 features to accurately predict a potential gross write rate. When the two most important features (NLSI score and age) are used to map a predicted gross write rate, the result is a multi-modal response surface containing numerous good combinations and numerous poor combinations. Figure 26 contains numerous peaks and valleys. The peaks represent higher GWR and the valleys are lower GWR. The peaks and valleys are scattered through the response surface. A NLSI cut off score or a NLSI range could be established if a high and large plateau existed on the surface. Unfortunately, such a plateau doesn't exist.



**Figure 26, Response Surface of Predicted Gross Write Rate as a Function of NLSI Score and Age**

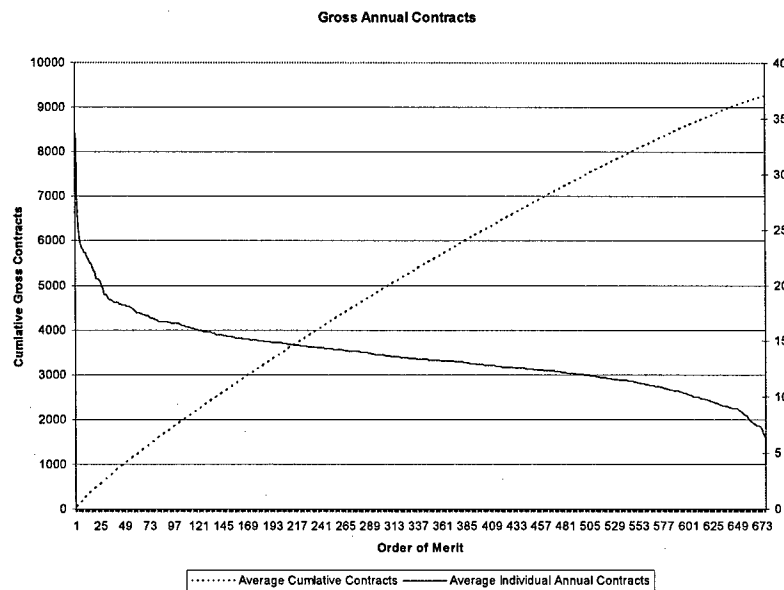
The model produces an order of merit (OML). The OML is extremely useful for selecting those individuals with the inherent skills required for recruiting. The predicted gross write rate measures the individual's potential gross write rate and is technically a statistical expectation. However, it is dependent upon recruiter placement, mission requirements, and local market conditions. Recruiter selection is a system within a more complex recruiting system. Even the best recruiter, who is positioned wrong, given the wrong mission, and placed in low penetration markets, could have difficulty meeting their statistical expectation. Ultimately, the model's success also relies upon sound recruiter position and mission for the market.

We have two concerns that require observation once the model is operational. Total systems effects should be monitored. Potential recruiter performance may not be reaped if the entire complex recruiting system is not performing well. An example of a diminished system effect is placing a potential high production recruiter into recruiting station with many good recruiters and a low total station mission. The mission coupled with recruiter positioning places a low ceiling on the recruiter's performance, restricting his or her potential. We should also

observe for Dr. Goldberg's research thesis. Dr. Goldberg's study suggests more recruiters increase Army awareness and may help increase gross write rate (Goldberg, 2003).

Given these two concerns, we recommend the model's accumulative predicted production not be optimized until we have observed increased performance by those initially selected by the OML. The OML does identify those with the inherent skills for recruiting, so we expect better command gross write rates within a few years. This selection system is much better than the current selection system. But without knowing the potential system effects measured by system effects and Dr. Goldberg's thesis, optimizing the OML sooner, rather than later, may be a risk seeking decision. A risk averse approach would be gradually applying the OML predicted accumulative production towards recruiter selection numbers. Until we are able to synchronize recruiter selection with mission, market, and recruiter placement, the OML should be enforced more as an art than a science.

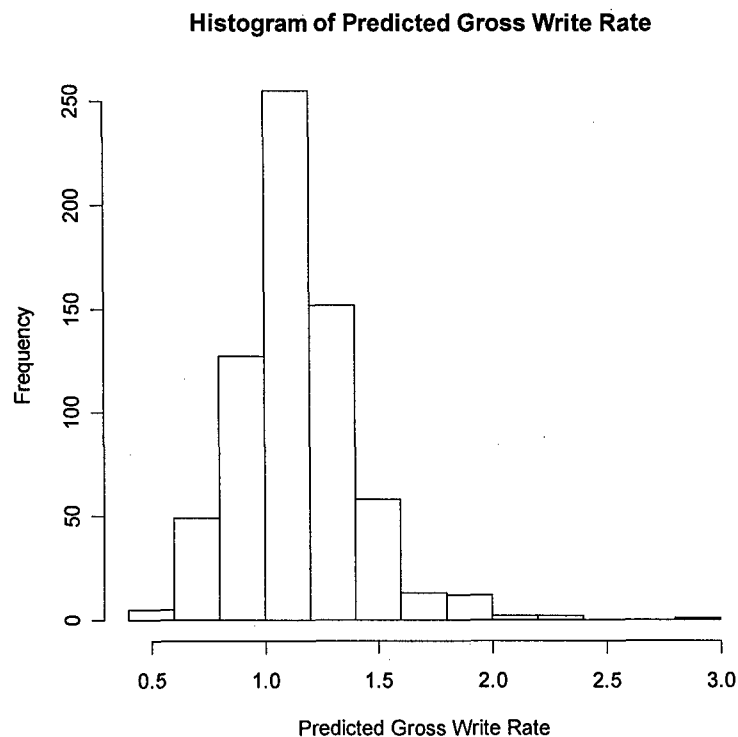
With regards to the model's output, the model produces numeric and graphical output. The model program ends on the graphical output, which plots accumulative predictive annual gross contracts on the left axis and individual predictive annual gross contracts. An example of the output is provided by Figure 27.



**Figure 27, Gross Annual Contracts Distributed by Order of Merit List (Individual Contracts are the solid line and Accumulative Contracts are the dashed lines)**



For this example, a frequency plot of the individual contracts would be approximately Gaussian. This is self evident by the majority of this sample containing annual contracts between sixteen and twelve contracts. The distribution is validated by Figure 28, which is the histogram of the predicted gross write rates provided by one model run. The tails of the distribution would contain either those who have potential to write more than sixteen contracts a year or those who potentially write less than twelve contracts a year. Applying lean six sigma philosophy, we would seek to obtain as many of those in the performance tail, the right side of Figure 28 or the left side of Figure 27. In application, the more we can test and classify, the more we can push into the performance tail away from the mean GWR. Pushing into the performance tail provides the ROI. A selected force has the potential to write more contracts in a given time specification. It follows that the more of the best performing recruiters we can select, the more contracts can be written and more NCO can be retained in the operational Army.



**Figure 28, Histogram of Model's Output, Predicted Gross Write Rate**

The numeric output provides a by social security OML. The OML contains: 1) the social security number; 2) the predicted GWR; 3) the average individual annual gross contracts; and 4)

an average accumulative annual contract amount (see Table 4). The model permits the cutting and pasting of the output into other database and spreadsheet applications. This facilitates the collection of many model runs, which can create a master OML. The master OML can easily be sorted by predicted GWR and the average cumulative contracts can be recalculated to establish potential selections.

**Table 4, Recruiter Order of Merit List Example**

ssn	predicted.gwr	Average Individual Annual Contracts	Average Cumulative Contracts
123456789	2.803469772	33.64163726	33.64163726
123456789	2.360912398	28.33094877	61.97258603
123456789	2.162760776	25.95312931	87.92571535
123456789	2.077832903	24.93399484	112.8597102
123456789	2.025033522	24.30040227	137.1601125
123456789	1.978600211	23.74320253	160.903315
123456789	1.945475518	23.34570621	184.2490212
123456789	1.936914642	23.2429757	207.4919969
123456789	1.915691164	22.98829397	230.4802909
123456789	1.912815964	22.95379157	253.4340824
123456789	1.909539949	22.91447939	276.3485618
123456789	1.887911847	22.65494217	299.003504
123456789	1.866458611	22.39750333	321.4010073
123456789	1.845946691	22.15136029	343.5523676
123456789	1.835997152	22.03196583	365.5843334
123456789	1.834655514	22.01586617	387.6001996
123456789	1.812446162	21.74935394	409.3495535
123456789	1.77619874	21.31438488	430.6639384
123456789	1.775705718	21.30846861	451.972407
123456789	1.742401321	20.90881586	472.8812229
123456789	1.715442747	20.58531296	493.4665359
123456789	1.715406372	20.58487646	514.0514123
123456789	1.713534577	20.56241492	534.6138272
123456789	1.704421789	20.45306146	555.0668887
123456789	1.688114859	20.25737831	575.324267
123456789	1.654195503	19.85034603	595.174613
123456789	1.654016601	19.84819921	615.0228123

## 5.2 Data Structures

### 5.2.1. Model Prediction Data

The input vector has the form of a tab delimited text file and should be saved as nlsi.txt. The data has to be ordered as: SSN, AGE, BIO001, BIO006, BIO007, BIO008, BIO014, BIO018, BIO022, BIO023, BIO033, BIO037, BIO038, BIO040, BIO041, BIO046, BIO049, BIO050, BIO054, BIO061, BIO064, BIO066, BIO071, BIO078, BIO087, BIO090, BIO093, BIO095, BIO102, BIO103, BIO107, BIO109, BIO110, BIO113, BIO115, BIO116, NLSIPVSC. If the data are not ordered and labeled as above, the model will crash. The observations should contain no blanks and strictly are obtained by NLSI. Blanks will also cause the model to crash or produce erroneous results. With the exception of the social security number, the NLSI variables and outcome are numeric.

### 5.2.2. Model Implementation and Updates

We strongly recommend a two phased model implementation. The first phase observes the performance of those selected by the model for at least a year, or two. Once the selection system is proven in the field, then we recommend implementing a master OML that significantly reduces selected recruiters and retains more junior leaders in the operational army. During phase one, those individuals who demonstrate better predicted production should be selected. However, the selection should not be made using the accumulative average annual contract number. This phase should select recruiters beyond a “cut line” to reduce risk.

As with all statistical learning models, the RandomForest and feature selection should be updated once a year to ensure the best possible prediction. If time and resources permit, both the features and the RandomForest should be updated. As a minimum, the RandomForest should be updated.

## **Chapter 6: Conclusions**

### ***6.1 Research Contributions***

The primary research contribution is enhancing industrial and organization (IO) psychology with statistical learning. The application of statistical learning advances the current body of IO psychology knowledge by linking a psychological inventory with actual performance metrics. The linkage provides a tangible result to human resource managers. The result can be used to better select the individuals possessing inherent skill sets for the right performance objectives. The partnership between IO psychology and statistical learning also contributes to statistical learning by advancing the application of statistical learning into psychology.

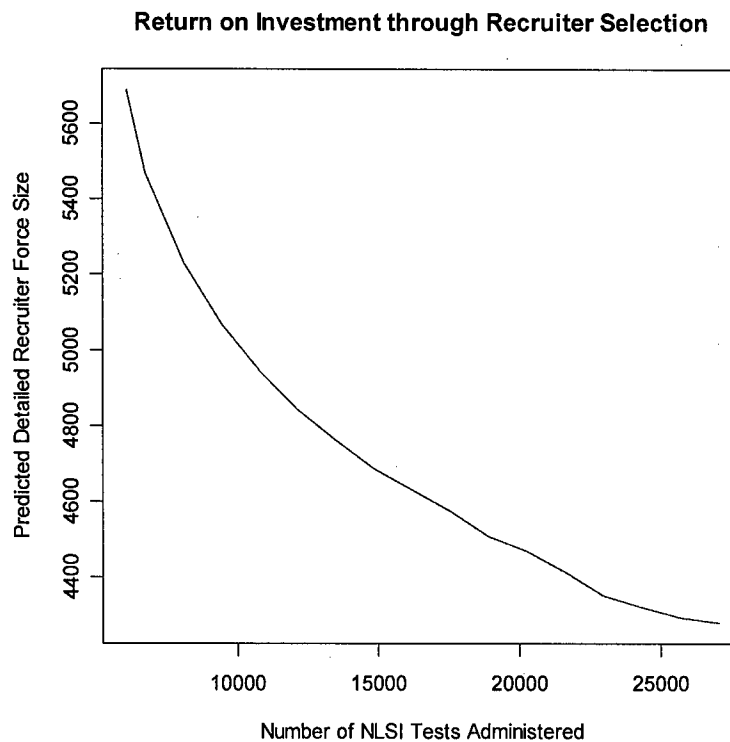
We also present a statistical learning methodology for determining the better combination of function approximation and features for prediction. The methodology iteratively converges on the function approximation paired with a set of features that produces the better prediction instrument. The iterative process uses a combination of multivariate statistics, feature selection methods, statistical learning, and analysis.

### ***6.2 Recommended Model Deployment***

We recommend that TRADOC maintain control of the model and HRC retain its current oversight on recruiter selection. The TRADOC manages the NCOES. Adding NLSI administration and NLSI score maintenance should not incur much additional cost; the marginal cost of adding an hour and a classroom to the current program of instruction is small. Because the Recruiter Selection Model is easily deployed from Excel, TRADOC can also maintain the OML and provide the OML to HRC. HRC retains its current recruiter selection oversight using the OML to enhance the selection.

### ***6.3 Estimated Return on Investment***

The estimation of return on investment is dependent upon the detailed recruiting force contract mission, the number of NCO given the NLSI test, and the model results. To estimate a return on investment, we assume: 1) a constant detailed recruiting force contract mission of 80,000 and 2) similar batch results provided by our test of 676 NCO. We increase the test population to demonstrate how an increase in test participants can reduce the size of the detailed recruiting force required to obtain the fixed number of 80K contracts.



**Figure 29, Return on Investment by Implementing NLSI and the Recruiter Selection Model**

Figure 29 demonstrates that by testing 250,000 NCO and using the recruiter selection model, the detailed recruiting force can be reduced from more than 5,600 to fewer than 4,400. Testing 250K reduces the detailed recruiting force by at least 1,200. Multiple costs are involved. Some of those costs include, but are not limited to, the cost per recruiter and the cost of NLSI administration. More difficult cost savings involve travel and school accounts and other transition costs. Depending on these individual costs, Recruiting Command can save the Army between \$65M to \$120M and significantly reduce junior leader transitions by retaining them in operational formations.

## Bibliography

- Baier, Thomas and Neuwirth, Erich. *RExcel – Using R from within Excel*. CRAN, 2006.
- Borman, Walter and Rosse, Rodney. An Empirical Construct Validity Approach to Study Predictor – Job Performance Links. *Journal of Applied Psychology*. 1980, Vol 65, No. 6, pp 662-671.
- Borman, Walter, et al. *U.S. Army Recruiter Selection Research: Development of the Non-Commissioned Officer Leadership Skills Inventory (NLSI)*. 46<sup>th</sup> Annual Conference of the International Military Testing Association, Brussels, Belgium, 2004.
- Brieman, Leo. *Random Forest – Random Features*. Technical Report 567. Statistics Department, University of California, Berkley, 1999.
- Brieman, Leo and Cutler, Adele. *The RandomForest Package, version 4.5-16*. CRAN, 2006.
- Brieman, Friedman, Olshen, and Stone. *Classification and Regression Trees*. Chapman & Hall, New York, 1998.
- Cristianini, Nello and Shawe-Taylor, John. *Support Vector Machines and Other Kernel Bases Learning Methods*. Cambridge University Press, United Kingdom, 2003.
- Dalgaard, Peter. *Introductory Statistics with R*. Springer, New York, 2002.
- Fox, John. *An R and S-Plus Companion to Applied Regression*. Sage Publications, Thousand Oaks, CA, 2002.
- Fukunaga, Kelnosuke. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 1990.
- George, Michael. *Lean Six-Sigma for Service*. McGraw-Hill, New York, 2003.
- George, Mike, Rowlands, Dave, and Kastle, Bill. *What is Lean Six Sigma?* McGraw-Hill, New York, 2004.
- George, Michael, Works, James, and Watson-Hemphill, Kimberly. *Fast Innovation*. McGraw-Hill, New York, 2005.
- Goldberg, Lawrence. *An Army Enlistment Early Warning System*. Institute for Defense Analysis, Alexandria, VA 2003.

- Hastie, Trevor, Tibshirani, Robert and Friedman, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2001.
- Halstead, John. *Support Vector Machine/Regression Feature Selection with an Application in Classification*. Doctoral Dissertation, University of Virginia, Charlottesville, VA, 2005.
- Halstead, John and Brown, Donald. *Improving Upon Logistic Regression to Reduce Army DEP Loss*. IEEE SIEDS Conference, Charlottesville, VA, 2004.
- Johnson, Richard and Wichern, Dean. *Applied Multivariate Statistical Analysis*. Prentice Hall, NJ, 2002.
- Kubisiak, Christean et al. *Concurrent Validation of the NLSI for U.S. Army Drill Sergeants*, Personal Decisions Research Institutes, Inc. Tampa, FL, June 2005.
- Neter, Kutner, Nachtsheim, and Wasserman. *Applied Linear Regression Models*. Irwin, Chicago, IL, 1996.
- Personnel Decisions Research Institute, Inc. *Overview of NLSI Research Program*. Presentation provided to Doctors Ross, Halstead, White, and Young. Tampa, FL, December 2005.
- Ross, Linda and Halstead, John. *NCO Leadership Skills Inventory (NLSI)*. Presentation provided to Major General Thomas Bostick, 9 January 2006.
- Scholkopf, Bernhard and Smola, Alexander. *Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, Cambridge, MA, 2002.
- Walkenbach, John. *Microsoft Excel 2000 Power Programming with VBA*. IDG Books Worldwide, Inc., Foster City, CA, 1999.
- Walkenbach, John. *Microsoft Excel 2003 Power Programming with VBA*. Wiley Publishing, Inc. Indianapolis, IN, 2004.

## Appendix A: List of Abbreviations

<b>A</b>	
ARC	Army Recruiting Course
ARI	Army Research Institute
AVF	All Volunteer Force
<b>C</b>	
CART	Classification and Regression Trees
<b>D</b>	
DTIC	Defense Technical Information Center
<b>G</b>	
GWR	Gross Write Rate
<b>H</b>	
HRC	Human Resource Command
<b>I</b>	
IO	Industrial and Organizational
<b>M</b>	
MOS	Military Occupational Skill
<b>N</b>	
NCO	Non Commissioned Officer
NCOES	Non Commissioned Officer Education System
NLSI	Non Commissioned Officer Leadership Skills Inventory
<b>O</b>	
ORCEN	Operations Research Center
<b>P</b>	
PDRI	Personnel Decisions Research Institute
<b>R</b>	
RRS	Recruiting and Retention School
<b>S</b>	
SE	Systems Engineering
SVR	Support Vector Regression
<b>T</b>	
TRADOC	Training and Doctrine Command
<b>U</b>	
USAAC	United States Army Accessions Command
USAREC	United States Army Recruiting Command
USMA	United States Military Academy



## Appendix B: Data Definitions

### NLSI Features

Feature Name	Feature Description
asgts	ASVAB general technical composite
bio001	Part I BIQ Raw Data - item 1
bio002	Part I BIQ Raw Data - item 2
bio003	Part I BIQ Raw Data - item 3
bio004	Part I BIQ Raw Data - item 4
bio005	Part I BIQ Raw Data - item 5
bio006	Part I BIQ Raw Data - item 6
bio007	Part I BIQ Raw Data - item 7
bio008	Part I BIQ Raw Data - item 8
bio009	Part I BIQ Raw Data - item 9
bio010	Part I BIQ Raw Data - item 10
bio011	Part I BIQ Raw Data - item 11
bio012	Part I BIQ Raw Data - item 12
bio013	Part I BIQ Raw Data - item 13
bio014	Part I BIQ Raw Data - item 14
bio015	Part I BIQ Raw Data - item 15
bio016	Part I BIQ Raw Data - item 16
bio017	Part I BIQ Raw Data - item 17
bio018	Part I BIQ Raw Data - item 18
bio019	Part I BIQ Raw Data - item 19
bio020	Part I BIQ Raw Data - item 20
bio021	Part I BIQ Raw Data - item 21
bio022	Part I BIQ Raw Data - item 22
bio023	Part I BIQ Raw Data - item 23
bio024	Part I BIQ Raw Data - item 24
bio025	Part I BIQ Raw Data - item 25
bio026	Part I BIQ Raw Data - item 26
bio027	Part I BIQ Raw Data - item 27
bio028	Part I BIQ Raw Data - item 28
bio029	Part I BIQ Raw Data - item 29
bio030	Part I BIQ Raw Data - item 30
bio031	Part I BIQ Raw Data - item 31
bio032	Part I BIQ Raw Data - item 32
bio033	Part I BIQ Raw Data - item 33
bio034	Part I BIQ Raw Data - item 34
bio035	Part I BIQ Raw Data - item 35
bio036	Part I BIQ Raw Data - item 36
bio037	Part I BIQ Raw Data - item 37
bio038	Part I BIQ Raw Data - item 38
bio039	Part I BIQ Raw Data - item 39
bio040	Part I BIQ Raw Data - item 40
bio041	Part I BIQ Raw Data - item 41
bio042	Part I BIQ Raw Data - item 42

bio043	Part I BIQ Raw Data - item 43
bio044	Part I BIQ Raw Data - item 44
bio045	Part I BIQ Raw Data - item 45
bio046	Part I BIQ Raw Data - item 46
bio047	Part I BIQ Raw Data - item 47
bio048	Part I BIQ Raw Data - item 48
bio049	Part I BIQ Raw Data - item 49
bio050	Part I BIQ Raw Data - item 50
bio051	Part I BIQ Raw Data - item 51
bio052	Part I BIQ Raw Data - item 52
bio053	Part I BIQ Raw Data - item 53
bio054	Part I BIQ Raw Data - item 54
bio055	Part I BIQ Raw Data - item 55
bio056	Part I BIQ Raw Data - item 56
bio057	Part I BIQ Raw Data - item 57
bio058	Part I BIQ Raw Data - item 58
bio059	Part I BIQ Raw Data - item 59
bio060	Part I BIQ Raw Data - item 60
bio061	Part I BIQ Raw Data - item 61
bio062	Part I BIQ Raw Data - item 62
bio063	Part I BIQ Raw Data - item 63
bio064	Part I BIQ Raw Data - item 64
bio065	Part I BIQ Raw Data - item 65
bio066	Part I BIQ Raw Data - item 66
bio067	Part I BIQ Raw Data - item 67
bio068	Part I BIQ Raw Data - item 68
bio069	Part I BIQ Raw Data - item 69
bio070	Part I BIQ Raw Data - item 70
bio071	Part I BIQ Raw Data - item 71
bio072	Part I BIQ Raw Data - item 72
bio073	Part I BIQ Raw Data - item 73
bio074	Part I BIQ Raw Data - item 74
bio075	Part I BIQ Raw Data - item 75
bio076	Part I BIQ Raw Data - item 76
bio077	Part I BIQ Raw Data - item 77
bio078	Part I BIQ Raw Data - item 78
bio079	Part I BIQ Raw Data - item 79
bio080	Part I BIQ Raw Data - item 80
bio081	Part I BIQ Raw Data - item 81
bio082	Part I BIQ Raw Data - item 82
bio083	Part I BIQ Raw Data - item 83
bio084	Part I BIQ Raw Data - item 84
bio085	Part I BIQ Raw Data - item 85
bio086	Part I BIQ Raw Data - item 86
bio087	Part I BIQ Raw Data - item 87
bio088	Part I BIQ Raw Data - item 88
bio089	Part I BIQ Raw Data - item 89
bio090	Part I BIQ Raw Data - item 90
bio091	Part I BIQ Raw Data - item 91

bio092	Part I BIQ Raw Data - item 92
bio093	Part I BIQ Raw Data - item 93
bio094	Part I BIQ Raw Data - item 94
bio095	Part I BIQ Raw Data - item 95
bio096	Part I BIQ Raw Data - item 96
bio097	Part I BIQ Raw Data - item 97
bio098	Part I BIQ Raw Data - item 98
bio099	Part I BIQ Raw Data - item 99
bio100	Part I BIQ Raw Data - item 100
bio101	Part I BIQ Raw Data - item 101
bio102	Part I BIQ Raw Data - item 102
bio103	Part I BIQ Raw Data - item 103
bio104	Part I BIQ Raw Data - item 104
bio105	Part I BIQ Raw Data - item 105
bio106	Part I BIQ Raw Data - item 106
bio107	Part I BIQ Raw Data - item 107
bio108	Part I BIQ Raw Data - item 108
bio109	Part I BIQ Raw Data - item 109
bio110	Part I BIQ Raw Data - item 110
bio111	Part I BIQ Raw Data - item 111
bio112	Part I BIQ Raw Data - item 112
bio113	Part I BIQ Raw Data - item 113
bio114	Part I BIQ Raw Data - item 114
bio115	Part I BIQ Raw Data - item 115
bio116	Part I BIQ Raw Data - item 116
bio117	Part I BIQ Raw Data - item 117
bio118	Part I BIQ Raw Data - item 118
bio119	Part I BIQ Raw Data - item 119
bio120	Part I BIQ Raw Data - item 120
bio121	Part I BIQ Raw Data - item 121
bio122	Part I BIQ Raw Data - item 122
bio123	Part I BIQ Raw Data - item 123
bio124	Part I BIQ Raw Data - item 124
bio125	Part I BIQ Raw Data - item 125
depend02	scored stem 02a for NLSI Part II 34 item form
agree02	scored stem 02b for NLSI Part II 34 item form
wrk02	scored stem 02c for NLSI Part II 34 item form
lead02	scored stem 02d for NLSI Part II 34 item form
adj03	scored stem 03a for NLSI Part II 34 item form
depend03	scored stem 03b for NLSI Part II 34 item form
wrk03	scored stem 03c for NLSI Part II 34 item form
lead03	scored stem 03d for NLSI Part II 34 item form
adj04	scored stem 04a for NLSI Part II 34 item form
pc04	scored stem 04b for NLSI Part II 34 item form
lead04	scored stem 04c for NLSI Part II 34 item form
wrk04	scored stem 04d for NLSI Part II 34 item form
pc05	scored stem 05a for NLSI Part II 34 item form
lead05	scored stem 05b for NLSI Part II 34 item form
adj05	scored stem 05c for NLSI Part II 34 item form

agree05	scored stem 05d for NLSI Part II 34 item form
adj06	scored stem 06a for NLSI Part II 34 item form
wrk06	scored stem 06b for NLSI Part II 34 item form
lie06	scored stem 06c for NLSI Part II 34 item form
agree06	scored stem 06d for NLSI Part II 34 item form
agree07	scored stem 07a for NLSI Part II 34 item form
lead07	scored stem 07b for NLSI Part II 34 item form
adj07	scored stem 07c for NLSI Part II 34 item form
wrk07	scored stem 07d for NLSI Part II 34 item form
pc08	scored stem 08a for NLSI Part II 34 item form
depend08	scored stem 08b for NLSI Part II 34 item form
wrk08	scored stem 08c for NLSI Part II 34 item form
lead08	scored stem 08d for NLSI Part II 34 item form
adj09	scored stem 09a for NLSI Part II 34 item form
wrk09	scored stem 09b for NLSI Part II 34 item form
agree09	scored stem 09c for NLSI Part II 34 item form
depend09	scored stem 09d for NLSI Part II 34 item form
lead10	scored stem 10a for NLSI Part II 34 item form
lie10	scored stem 10b for NLSI Part II 34 item form
agree10	scored stem 10c for NLSI Part II 34 item form
wrk10	scored stem 10d for NLSI Part II 34 item form
wrk11	scored stem 11a for NLSI Part II 34 item form
depend11	scored stem 11b for NLSI Part II 34 item form
adj11	scored stem 11c for NLSI Part II 34 item form
agree11	scored stem 11d for NLSI Part II 34 item form
lead12	scored stem 12a for NLSI Part II 34 item form
wrk12	scored stem 12b for NLSI Part II 34 item form
depend12	scored stem 12c for NLSI Part II 34 item form
adj12	scored stem 12d for NLSI Part II 34 item form
depend13	scored stem 13a for NLSI Part II 34 item form
lead13	scored stem 13b for NLSI Part II 34 item form
lie13	scored stem 13c for NLSI Part II 34 item form
wrk13	scored stem 13d for NLSI Part II 34 item form
depend14	scored stem 14a for NLSI Part II 34 item form
adj14	scored stem 14b for NLSI Part II 34 item form
agree14	scored stem 14c for NLSI Part II 34 item form
lead14	scored stem 14d for NLSI Part II 34 item form
lead15	scored stem 15a for NLSI Part II 34 item form
wrk15	scored stem 15b for NLSI Part II 34 item form
adj15	scored stem 15c for NLSI Part II 34 item form
pc15	scored stem 15d for NLSI Part II 34 item form
lie16	scored stem 16a for NLSI Part II 34 item form
wrk16	scored stem 16b for NLSI Part II 34 item form
agree16	scored stem 16c for NLSI Part II 34 item form
depend16	scored stem 16d for NLSI Part II 34 item form
wrk17	scored stem 17a for NLSI Part II 34 item form
adj17	scored stem 17b for NLSI Part II 34 item form
agree17	scored stem 17c for NLSI Part II 34 item form
dep17	scored stem 17d for NLSI Part II 34 item form

agree18	scored stem 18a for NLSI Part II 34 item form
depend18	scored stem 18b for NLSI Part II 34 item form
adj18	scored stem 18c for NLSI Part II 34 item form
wrk18	scored stem 18d for NLSI Part II 34 item form
adj19	scored stem 19a for NLSI Part II 34 item form
lie19	scored stem 19b for NLSI Part II 34 item form
agree19	scored stem 19c for NLSI Part II 34 item form
depend19	scored stem 19d for NLSI Part II 34 item form
wrk20	scored stem 20a for NLSI Part II 34 item form
depend20	scored stem 20b for NLSI Part II 34 item form
lead20	scored stem 20c for NLSI Part II 34 item form
adj20	scored stem 20d for NLSI Part II 34 item form
wrk21	scored stem 21a for NLSI Part II 34 item form
pc21	scored stem 21b for NLSI Part II 34 item form
lead21	scored stem 21c for NLSI Part II 34 item form
adj21	scored stem 21d for NLSI Part II 34 item form
lie22	scored stem 22a for NLSI Part II 34 item form
adj22	scored stem 22b for NLSI Part II 34 item form
agree22	scored stem 22c for NLSI Part II 34 item form
depend22	scored stem 22d for NLSI Part II 34 item form
adj23	scored stem 23a for NLSI Part II 34 item form
depend23	scored stem 23b for NLSI Part II 34 item form
lead23	scored stem 23c for NLSI Part II 34 item form
agree23	scored stem 23d for NLSI Part II 34 item form
agree24	scored stem 24a for NLSI Part II 34 item form
depend24	scored stem 24b for NLSI Part II 34 item form
adj24	scored stem 24c for NLSI Part II 34 item form
lead24	scored stem 24d for NLSI Part II 34 item form
lie25	scored stem 25a for NLSI Part II 34 item form
lead25	scored stem 25b for NLSI Part II 34 item form
agree25	scored stem 25c for NLSI Part II 34 item form
depend25	scored stem 25d for NLSI Part II 34 item form
agree26	scored stem 26a for NLSI Part II 34 item form
pc26	scored stem 26b for NLSI Part II 34 item form
depend26	scored stem 26c for NLSI Part II 34 item form
wrk26	scored stem 26d for NLSI Part II 34 item form
adj27	scored stem 27a for NLSI Part II 34 item form
depend27	scored stem 27b for NLSI Part II 34 item form
lead27	scored stem 27c for NLSI Part II 34 item form
pc27	scored stem 27d for NLSI Part II 34 item form
depend28	scored stem 28a for NLSI Part II 34 item form
lead28	scored stem 28b for NLSI Part II 34 item form
agree28	scored stem 28c for NLSI Part II 34 item form
pc28	scored stem 28d for NLSI Part II 34 item form
adj29	scored stem 29a for NLSI Part II 34 item form
lead29	scored stem 29b for NLSI Part II 34 item form
wrk29	scored stem 29c for NLSI Part II 34 item form
depend29	scored stem 29d for NLSI Part II 34 item form
pc30	scored stem 30a for NLSI Part II 34 item form

lead30	scored stem 30b for NLSI Part II 34 item form
wrk30	scored stem 30c for NLSI Part II 34 item form
adj30	scored stem 30d for NLSI Part II 34 item form
lead31	scored stem 31a for NLSI Part II 34 item form
lie31	scored stem 31b for NLSI Part II 34 item form
agree31	scored stem 31c for NLSI Part II 34 item form
wrk31	scored stem 31d for NLSI Part II 34 item form
wrk32	scored stem 32a for NLSI Part II 34 item form
pc32	scored stem 32b for NLSI Part II 34 item form
lead32	scored stem 32c for NLSI Part II 34 item form
adj32	scored stem 32d for NLSI Part II 34 item form
depend33	scored stem 33a for NLSI Part II 34 item form
adj33	scored stem 33b for NLSI Part II 34 item form
agree33	scored stem 33c for NLSI Part II 34 item form
wrk33	scored stem 33d for NLSI Part II 34 item form
wrk34	scored stem 34a for NLSI Part II 34 item form
agree34	scored stem 34b for NLSI Part II 34 item form
lead34	scored stem 34c for NLSI Part II 34 item form
adj34	scored stem 34d for NLSI Part II 34 item form
nlsipvsc	NLSI PV Keyed Score= 50 + (nlsipvky *10)
grs_avg6	Gross Write Rate

---

## Appendix C: R Code

### Recruiter Selection Code

```
# Load data and libraries #####
# w/o nlsi score #####

train <- read.table ("c:/data/recruiter/nlsi_training.txt", header=T, fill=T)
val <- read.table ("c:/data/recruiter/nlsi_val.txt", header=T, fill=T)

library(stats)
library(graphics)
library(MASS)
library(tree)
library(lattice)
library(randomForest)
library(RColorBrewer)
library(car)
library(class)
library(e1071)

# statistics #####

mu <- mean(train)
sigma2 <- var(train)
Corr <-cor(train)

# Random Forest ###
# Provides important variables ###

RF <- randomForest(GRS_AVG6 ~ ., data=train, na.action=na.omit)
RF
round(importance(RF), 2)
imp <- round(importance(RF), 2)
imp.o <- order(imp, decreasing = TRUE)
newtrain.RF <- train[imp.o]
important.RF <- colnames(newtrain.RF)
important.RF
```

```
summary (RF)
```

```
# Variable selection by Marlow Cp = AIC with Regression #####
```

```
Model.nlsi <- lm(GRS_AVG6 ~ ., data=train, na.action=na.omit)
nlsi.step <- step(Model.nlsi, direction = "both")
```

```
summary(nlsi.step)
```

```
anova(nlsi.step)
```

```
# Analysis incorporating feature selection from above #####
```

```
# Load data and libraries #####
```

```
library(stats)
```

```
library(graphics)
```

```
library(MASS)
```

```
library(tree)
```

```
library(lattice)
```

```
library(randomForest)
```

```
library(RColorBrewer)
```

```
library(car)
```

```
library(class)
```

```
library(e1071)
```

```
T1 <- read.table ("c:/data/recruiter/reg_train.txt", header=T)
```

```
V1 <- read.table ("c:/data/recruiter/reg_val.txt", header=T)
```

```
T2 <- read.table ("c:/data/recruiter/rf_train.txt", header=T)
```

```
V2 <- read.table ("c:/data/recruiter/rf_val.txt", header=T)
```

```
Comp <- read.table("c:/data/recruiter/score_corr.txt", header=T)
```

```
plot(Comp[,1],Comp[,2], xlab = "NLSI Score", ylab = "Gross Write Rate (minus first six months)" )
```

```
Nlsi.cor <- cor(Comp[,1], Comp[,2])
```

```
cor.title <- paste("Correlation between NLSI Score and Gross Write Rate (Corr =",
",paste(round(Nlsi.cor,4)),")", sep = "")
```

```
title(cor.title)
```



```
# RF modeling #####
```

```
RF.1 <- randomForest(GRS_AVG6 ~ ., data=T2)
```

```
RF.1
```

```
round(importance(RF.1), 2)
```

```
imp <- round(importance(RF.1), 2)
```

```
imp.o <- order(imp, decreasing = TRUE)
```

```
newtrain.RF.1 <- T2[imp.o]
```

```
important.RF.1 <- colnames(newtrain.RF.1)
```

```
important.RF.1
```

```
summary (RF.1)
```

```
plot(RF.1, main="Random Forest Error vs. Number of Trees")
```

```
varImpPlot(RF.1, n.var=10, main= "Random Forest Important Variables")
```

```
RF.2 <- predict (RF.1, newdata = V2)
```

```
Y.RF <- V2[,ncol(V2)]
```

```
SSE.RF <- sum((Y.RF-RF.2)^2)
```

```
SSE.RF
```

```
MSE.RF <- SSE.RF/nrow(V2)
```

```
MSE.RF
```

```
plot(V2[, (ncol(V2)-1)], V2[, ncol(V2)], xlab = "NLSI Score", ylab = "Gross Write Rate", main = "Random Forest Model")
```

```
points(V2[, (ncol(V2)-1)], RF.2, col = 3)
```

```
legend(locator(n=1), legend=c("Actual Data", "Random Forest Prediction"), col = c(1, 3), lty = c(1,1))
```

```
rf.sub <- paste("MSE = ", paste(round(MSE.RF, 6)), sep = "")
```

```
text(locator(n=1), rf.sub)
```

```
# CART Regression Tree Modeling #####
```

```
Tree.1 <- tree(GRS_AVG6 ~ ., T2)
```

```
Tree.1
```

```
summary(Tree.1)
```

```
plot(Tree.1)
title("Regression Tree for Predicting Gross Write Rate")
text(Tree.1)
```

```
### Regression model #####
```

```
rm <- lm(GRS_AVG6 ~ ., data=T1)
rms <- step(rm, direction = "both")
```

```
summary(rms)
anova(rms)
```

```
rms.v <- predict(rms, newdata = V1)
Y.rms <- V1[,ncol(V1)]
SSE.rms <- sum((Y.rms-rms.v)^2)
SSE.rms
MSE.rms <- SSE.rms/nrow(V1)
MSE.rms
```

```
par(mfrow=c(2,2), mex=0.6)
plot(rms)
par(mfrow=c(1,1), mex=1)
```

```
plot(V1[, (ncol(V1)-1)], V1[, ncol(V1)], xlab = "NLSI Score", ylab = "Gross Write Rate", main = "Step Wise
Regression Model")
points(V1[, (ncol(V1)-1)], rms.v, col = 6)
legend(locator(n=1), legend=c("Actual Data", "Step Regression Prediction"), col = c(1, 6), lty = c(1,1))
rms.sub <- paste("MSE = ", paste(round(MSE.rms,6)), sep = "")
text(locator(n=1), rms.sub)
```

```
### SVR modeling #####
```

```
Svr1 <- svm(GRS_AVG6 ~ ., data=T1)
Svr2 <- svm(GRS_AVG6 ~ ., data=T2)
summary(Svr1)
summary(Svr2)
```

```
plot(Svr1)
```

```
svr1.val <- predict(Svr1, newdata=V1)
```

```
svr2.val <- predict(Svr2, newdata=V2)
```

```
Y.1 <- V1[,ncol(V1)]
```

```
SSE.svr1 <- sum((Y.1-svr1.val)^2)
```

```
SSE.svr1
```

```
MSE.svr1 <- SSE.svr1/nrow(V1)
```

```
MSE.svr1
```

```
Y.2 <- V2[,ncol(V2)]
```

```
SSE.svr2 <- sum((Y.2-svr2.val)^2)
```

```
SSE.svr2
```

```
MSE.svr2 <- SSE.svr2/nrow(V2)
```

```
MSE.svr2
```

```
plot(V1[, (ncol(V1)-1)], V1[, ncol(V1)], xlab = "NLSI Score", ylab = "Gross Write Rate", main = "Support  
Vector Regression Model 1")
```

```
points(V1[, (ncol(V1)-1)], svr1.val, col = 4)
```

```
legend(locator(n=1), legend=c("Actual Data", "Support Vector Prediction"), col = c(1, 4), lty = c(1, 1))
```

```
svr1.sub <- paste("MSE = ", paste(round(MSE.svr1, 6)), sep = "")
```

```
text(locator(n=1), svr1.sub)
```

```
plot(V2[, (ncol(V2)-1)], V2[, ncol(V2)], xlab = "NLSI Score", ylab = "Gross Write Rate", main = "Support  
Vector Regression Model 2")
```

```
points(V2[, (ncol(V2)-1)], svr2.val, col = 2)
```

```
legend(locator(n=1), legend=c("Actual Data", "Support Vector Prediction"), col = c(1, 2), lty = c(1, 1))
```

```
svr2.sub <- paste("MSE = ", paste(round(MSE.svr2, 6)), sep = "")
```

```
text(locator(n=1), svr2.sub)
```

```
# Data visualization for important variables ####
```

```
library(stats)
```

```
library(graphics)
```

```
library(MASS)
```

```
library(tree)
```

```
library(lattice)
```

```
library(randomForest)
```

```
library(RColorBrewer)
```

```
library(car)
```

```
library(class)
```

```
library(e1071)
```

```
data.vis <- read.table ("c:/data/recruiter/rf_imp_features.txt", header=T)
```

```
par(mex=0.5)
```

```
pairs(data.vis, gap=0, cex.labels=1.0, main="Pairwise Scatter Plots of Important Features")
```

```
imp.cor <- cor(data.vis)
```

### **Random Forest Greedy Algorithm to Determine Best Feature Subset**

```
# Load data and libraries #####
```

```
library(stats)
```

```
library(graphics)
```

```
library(MASS)
```

```
library(tree)
```

```
library(lattice)
```

```
library(randomForest)
```

```
library(RColorBrewer)
```

```
library(car)
```

```
library(class)
```

```
T2 <- read.table ("c:/data/recruiter/rf_train.txt", header=T)
```

```
V2 <- read.table ("c:/data/recruiter/rf_val.txt", header=T)
```

```
Comp <- read.table("c:/data/recruiter/score_corr.txt", header=T)
```

```
windows()
```

```
plot(Comp[,1],Comp[,2], xlab = "NLSI Score", ylab = "Gross Write Rate (minus first six months)" )
```

```
Nlsi.cor <- cor(Comp[,1], Comp[,2])
```

```
cor.title <- paste("Correlation between NLSI Score and Gross Write Rate (Corr =", paste(round(Nlsi.cor,4)),")", sep = "")
```

```
title(cor.title)
```

```
# RF modeling #####
```

```
RF.1 <- randomForest(GRS_AVG6 ~ ., data=T2)
```

```
RF.1
```

```
round(importance(RF.1), 2)
```

```
imp <- round(importance(RF.1), 2)
```

```
imp.o <- order(imp, decreasing = TRUE)
```

```
newtrain.RF.1 <- T2[imp.o]
```

```
important.RF.1 <- colnames(newtrain.RF.1)
```

```
important.RF.1
```

```
summary (RF.1)
```

```
windows()
```

```
plot(RF.1, main="Random Forest Error vs. Number of Trees")
```

```
windows()
```

```
varImpPlot(RF.1, n.var=36, main= "Random Forest Important Variables")
```

```
RF.2 <- predict (RF.1, newdata = V2)
```

```
Y.RF <- V2[,ncol(V2)]
```

```
SSE.RF <- sum((Y.RF-RF.2)^2)
```

```
SSE.RF
```

```
MSE.RF <- SSE.RF/nrow(V2)
```

```
MSE.RF
```

```
windows()
```

```
plot(V2[, (ncol(V2)-1)], V2[, ncol(V2)], xlab = "NLSI Score", ylab = "Gross Write Rate", main = "Random Forest Model")
```

```
points(V2[, (ncol(V2)-1)], RF.2, col = 3)
```

```

legend(locator(n=1), legend=c("Actual Data", "Random Forest Prediction"), col = c(1, 3), lty = c(1,1))
rf.sub <- paste("MSE = ",paste(round(MSE.RF,6)), sep = "")
text(locator(n=1), rf.sub)

```

```

# Greedy Algorithm

```

```

# Goal is to record the MSE of each RF model built from subsets of good best predictor variables

```

```

#

```

```

newval.RF.1 <- V2[imp.o]

```

```

score <- NULL

```

```

model.data <- NULL

```

```

val.data <- NULL

```

```

model.x <- NULL

```

```

val.x <- NULL

```

```

model.y <- T2[ncol(T2)]

```

```

val.y <- V2[ncol(V2)]

```

```

for (j in 1:ncol(newtrain.RF.1)){

```

```

  if (j ==1){

```

```

    model.x <- newtrain.RF.1[j]

```

```

    val.x <- newval.RF.1[j]

```

```

    model.data <- cbind(model.y, model.x)

```

```

    val.data <- cbind(val.y, val.x)

```

```

    loop.rf <- randomForest(GRS_AVG6 ~ ., data=model.data)

```

```

    loop.rf.v <- predict (loop.rf, newdata = val.data)

```

```

    SSE.RF <- sum((val.y-loop.rf.v)^2)

```

```

    MSE.RF <- SSE.RF/nrow(V2)

```

```

    score <- c(score, MSE.RF, j)

```

```

  }

```

```

  if (j > 1){

```

```

    model.x <- cbind(model.x, newtrain.RF.1[j])

```

```

    val.x <- cbind(val.x, newval.RF.1[j])

```

```

    model.data <- cbind(model.y, model.x)

```

```

    val.data <- cbind(val.y, val.x)

```

```

    loop.rf <- randomForest(GRS_AVG6 ~ ., data=model.data)

```

```

    loop.rf.v <- predict (loop.rf, newdata = val.data)

```

```

    SSE.RF <- sum((val.y-loop.rf.v)^2)

```

```

    MSE.RF <- SSE.RF/nrow(V2)

```

```

    score <- c(score, MSE.RF, j)

```

```

    }
  }

score
vis.data <- matrix(score, nrow=j, byrow=T)
colnames(vis.data) <- c("mse", "subset_size")
best.mse <- vis.data[vis.data[,1]<=min(vis.data[,1])]

windows()
plot(vis.data[,2], vis.data[,1], type = "l", col = 4, xlab="Subset Size", ylab="MSE", main= "Random Forest
Best Set of Ordered Important Variables" )
legend(locator(n=1), legend=c("Best MSE =", paste(best.mse[1]), "Best Subset Size =",
paste(best.mse[2])))

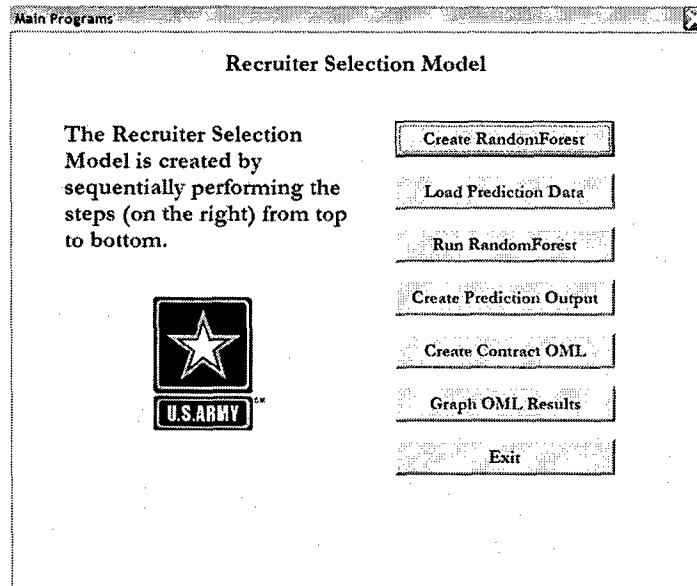
# best model

model.x <- newtrain.RF.1[,1:36]
val.x <- newval.RF.1[,1:36]
model.data <- cbind(model.y, model.x)
val.data <- cbind(val.y, val.x)
best.rf <- randomForest(GRS_AVG6 ~ ., data=model.data)
best.rf.v <- predict (best.rf, newdata = val.data)
mse.score <- round(best.mse[1], 6)

windows()
plot(val.data[,3], val.data[,1], xlab = "NLSI Score", ylab = "Gross Write Rate", main = "Best Random
Forest Model")
points(val.data[,3], best.rf.v, col = 3)
legend(locator(n=1), legend=c("Actual Data", "Best Random Forest Prediction"), col = c(1, 3), lty = c(1,1))
rf.sub <- paste("MSE = ",paste(mse.score), sep = "")
text(locator(n=1), rf.sub)

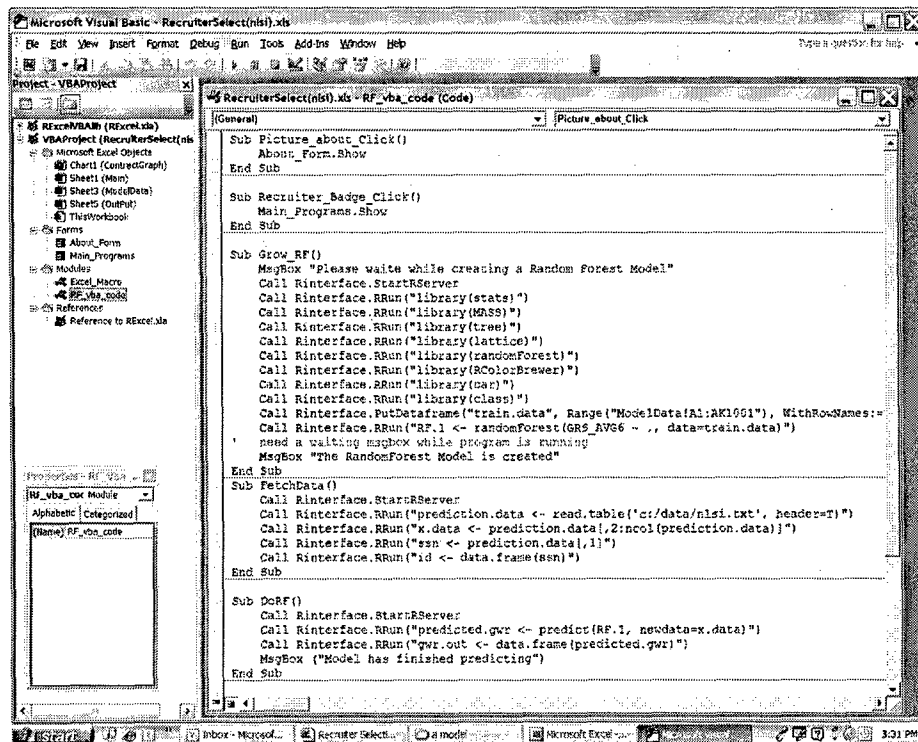
```

## Appendix D: RExcel Random Forest Model



Recruiter Selection Model RExcel Programs:

### VBA Objects:



### RF vba code

Sub Picture\_about\_Click()



```
About_Form.Show
```

```
End Sub
```

```
Sub Recruiter_Badge_Click()
```

```
    Main_Programs.Show
```

```
End Sub
```

```
Sub Grow_RF()
```

```
    MsgBox "Please wait while creating a Random Forest Model"
```

```
    Call Rinterface.StartRServer
```

```
    Call Rinterface.RRun("library(stats)")
```

```
    Call Rinterface.RRun("library(MASS)")
```

```
    Call Rinterface.RRun("library(tree)")
```

```
    Call Rinterface.RRun("library(lattice)")
```

```
    Call Rinterface.RRun("library(randomForest)")
```

```
    Call Rinterface.RRun("library(RColorBrewer)")
```

```
    Call Rinterface.RRun("library(car)")
```

```
    Call Rinterface.RRun("library(class)")
```

```
    Call Rinterface.PutDataframe("train.data", Range("ModelData!A1:AK1001"),  
    WithRowNames:=False)
```

```
    Call Rinterface.RRun("RF.1 <- randomForest(GRS_AVG6 ~ ., data=train.data)")
```

```
    ' need a waiting msgbox while program is running
```

```
    MsgBox "The RandomForest Model is created"
```

```
End Sub
```

```
Sub FetchData()
```

```
    Call Rinterface.StartRServer
```

```
    Call Rinterface.RRun("prediction.data <- read.table('c:/data/nlsi.txt', header=T)")
```

```
    Call Rinterface.RRun("x.data <- prediction.data[,2:ncol(prediction.data)]")
```

```
    Call Rinterface.RRun("ssn <- prediction.data[,1]")
```

```
    Call Rinterface.RRun("id <- data.frame(ssn)")
```

```
End Sub
```

Sub DoRF()

Call Rinterface.StartRServer

Call Rinterface.RRun("predicted.gwr <- predict(RF.1, newdata=x.data)")

Call Rinterface.RRun("gwr.out <- data.frame(predicted.gwr)")

MsgBox ("Model has finished predicting")

End Sub

Sub DisplayOut()

Call Rinterface.StartRServer

Call Rinterface.GetDataframe("id", Range("OutPut!A1"))

Call Rinterface.RRun("check <- is (gwr.out, 'data.frame')")

Call Rinterface.RRun("while(check==FALSE){gwr.out<-data.frame(predicted.gwr)}")

Call Rinterface.GetDataframe("gwr.out", Range("OutPut!B1"))

MsgBox ("Predicted GWR is in OutPut Worksheet")

Worksheets("OutPut").Activate

End Sub

### **Excel Macro Code:**

Sub OML\_Contracts()

Columns("A:B").Select

Selection.Sort Key1:=Range("B2"), Order1:=xlDescending, Header:=xlGuess, \_

OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom, \_

DataOption1:=xlSortNormal

Range("C2").Select

ActiveCell.FormulaR1C1 = "=12\*RC[-1]"

Range("C2").Select

Selection.AutoFill Destination:=Range("C2:C677")

Range("C2:C677").Select

Range("D2").Select

ActiveCell.FormulaR1C1 = "=RC[-1]"

```
Range("D3").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+RC[-1]"
Range("D3").Select
Selection.AutoFill Destination:=Range("D3:D677")
Range("D3:D677").Select
MsgBox ("Check out the graph")
End Sub
```

## Distribution List

The list indicates the complete mailing address of the individuals and organizations receiving copies of the report and the number of copies received. Due to the Privacy Act, only use business addresses; no personal home addresses. Distribution lists provide a permanent record of initial distribution. The distribution information will include the following entries:

NAME/AGENCY		ADDRESS	COPIES
Author(s)		Department of Systems Engineering Mahan Hall West Point, NY 10996	5
Client		U.S. Army Accessions Command Center for Accessions Research Fort Knox, KY 40121-2725	3
Dean, USMA		Office of the Dean Building 600 West Point, NY 10996	1
Defense Information (DTIC)	Technical Center	ATTN: DTIC-O Defense Technical Information Center 8725 John J. Kingman Rd, Suite 0944 Fort Belvoir, VA 22060-6218	1
Department Head-DSE		Department of Systems Engineering Mahan Hall West Point, NY 10996	1
ORCEN		Department of Systems Engineering Mahan Hall West Point, NY 10996	5
ORCEN Director		Department of Systems Engineering Mahan Hall West Point, NY 10996	1
USMA Library		USMA Library Bldg 757 West Point, NY 10996	1

# REPORT DOCUMENTATION PAGE – SF298

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> 10-03-2006		<b>2. REPORT TYPE</b> Technical Report	<b>3. DATES COVERED (From - To)</b> Dec 2005 to Mar 2006		
<b>4. TITLE AND SUBTITLE</b>  Recruiter Selection Model			<b>5a. CONTRACT NUMBER</b>		
			<b>5b. GRANT NUMBER</b>		
			<b>5c. PROGRAM ELEMENT NUMBER</b>		
<b>6. AUTHOR(S)</b>  John Brantley Halstead			<b>5d. PROJECT NUMBER</b>		
			<b>5e. TASK NUMBER</b>		
			<b>5f. WORK UNIT NUMBER</b>		
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  Operations Research Center US Military Academy 646 Swift Road West Point, NY 10996			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  DSE-TR-0623		
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  U.S. Army Accessions Command Center for Accessions Fort Knox, KY 40121-2725			<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>		
			<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>		
<b>12. DISTRIBUTION / AVAILABILITY STATEMENT</b> Distribution A: Approved for public release; distribution is unlimited					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b> This research provides statistical prediction of job performance derived from psychological inventories and biographical data. The research uses a combination of statistical learning, feature selection methods, and multivariate statistics to determine the better prediction function approximation with features obtained from the Non Commissioned Officer Leadership Skills Inventory (NLSI) and biographical data. The research created a methodology for iteratively developing a statistical learning model. The resulting RandomForest model runs in R statistical language and is controlled within an Excel worksheet environment by using Visual Basic Application (VBA) language to call R. The model provides significant cost benefits to the Army. Implementation of the model assists the Army with providing the right NCO to recruiting and the operational Army.					
<b>15. SUBJECT TERMS</b> Statistical Learning, Feature Selection, NLSI, Prediction, Regression, Random Forest					
<b>16. SECURITY CLASSIFICATION OF:</b> Unclassified			<b>17. LIMITATION OF ABSTRACT</b> Unclassified	<b>18. NUMBER OF PAGES</b>  66	<b>19a. NAME OF RESPONSIBLE PERSON</b> LTC Simon R. Goerger, PhD
<b>a. REPORT</b> Unclassified	<b>b. ABSTRACT</b> Unclassified	<b>c. THIS PAGE</b> Unclassified			<b>19b. TELEPHONE NUMBER (include area code)</b> 845-938-5897

Standard Form 298 (Rev. 8-98)  
Prescribed by ANSI Std. Z39.18